(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 720 413 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
03.07.1996 Bulletin 1996/27

(51) Int. Cl.[6]: **H04Q 11/04**

(21) Application number: **95309013.1**

(22) Date of filing: **12.12.1995**

(84) Designated Contracting States:
BE DE ES FR GB IT NL

(30) Priority: **30.12.1994 US 366704**
30.12.1994 US 367489
30.12.1994 US 366707
30.12.1994 US 366708

(71) Applicant **AT&T Corp.**
New York, NY 10013-2412 (US)

(72) Inventors:
- **Cloonan, Thomas Jay**
  Downers Grove, Illinois 60516 (US)
- **Richards, Gaylord Warner**
  Lisle, Illinois 60532 (US)

(74) Representative: **Watts, Christopher Malcolm Kelway, Dr. et al**
Lucent Technologies (UK) Ltd,
5 Mornington Road
Woodford Green Essex, IG8 0TU (GB)

(54) Terabit per second packet switch

(57) A physically realizable one terabit or more ATM packet switch 10A that has a large number of input interfaces connected to a single stage switching fabric 14A which is in turn connected to a number of output modules 16_0-16_m , according to the generic growable packet switch architecture. This ATM packet switch 10A is different from other growable packet switches in that it has a single stage switch fabric (14A or 14B) controlled by an out-of-band controller 20, yet it has significantly reduced complexity with respect to comparably sized electronic crossbar switches or their isomorphs. This ATM packet switch architecture is so flexible, it can be extended to provide variable length packets, circuit switched connections and fault tolerant redundant circuits using the same switch fabric and out-of-band controller architectures.

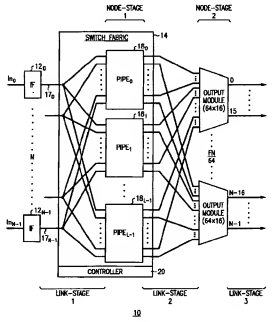


FIG. 3

EP 0 720 413 A2

Description

Technical Field

The invention relates to large telecommunication switches and more particularly to large telecommunication switches that use data packets in order to communicate at aggregate throughputs at the one terabit per second level.

Description of the Prior Art

Telecommunications have long used digital switching to encode, multiplex, transmit and decode audio frequencies in order to carry the millions of telephone voice calls of the world. Telecommunication switches for voice calls have grown to very large sizes to keep pace with demand. Most of the switching systems that route and control voice call traffic are called circuit switches, which means that for each call a type of bi-directional circuit, usually audio is set up between the calling party and the called parry. The circuit that is set up has the bandwidth and transport timing necessary to simulate a face-to-face conversation without distortion or time delays that are objectionable to the parties.

An alternative to circuit switching is called packet switching. For packet switching, the calling parry is responsible for converting the information into one or more packets. This information could be encoded voice, it could be encoded computer data, or it could be encoded video. A number to guide the packet to its destination, i.e. the called party, is typically included in a packet header. The packet switching network then has the task of routing each packet to its respective destination without undue distortion or delay. The called party usually has the equipment to receive the packets and decode the information back into an appropriate form.

The extremely rapid growth of packet switching traffic carrying voice, computer (LAN/WAN) , facsimile, image and video data to an ever widening variety of locations, along with the proposals of the Information Infrastructure, has challenged both the present packet switch system architectures and protocols.

Many vendors and service providers have joined forces to define a global standard that permits packet switching services to be provided in a ubiquitous fashion. The result of this coordinated effort by the industry has been the rapid development and deployment of an Asynchronous Transfer Mode (ATM) as a standard means of efficiently routing and transporting data packets that have stochastically-distributed arrival rates according to the recent ATM standard. ATM is thus a packet-oriented standard, but unlike most of its data packet predecessors (X.25, frame relay, etc.), ATM uses short, fixed-length, 53-byte packets that are also known as cells. ATM also uses a very streamlined form of error recovery and flow control relative to its predecessors. In fact, the ATM standard essentially eliminates most error protection and flow control at the link level, leaving these functions to higher level protocols at the edges of the network. This approach permits rapid routing of the short cells with minimal network delay and jitter, making ATM compatible with voice, data and video services. ATM has been embraced by the computer, LAN, and WAN industries, so a seamless packet communication from the source computer through LANs, WANs, and the public-switched network is already a reality.

If this level of connectivity becomes available to the average consumer and if advanced broad band services that combine voice, broadband data and video are similarly available at reasonable prices, then the volume of ATM traffic that may be generated in the future is virtually limitless. As a result, the number and size of the switches and cross-connects required to route the projected ATM packet traffic may also grow by phenomenal rates within the next decade. ATM switches and cross-connects for toll and gateway applications may require aggregate bandwidths ranging from 155 gigabits per second (1000 inputs at SONET OC-3 155 Mbps rates) to 2.4 terabits per second (1000 inputs at SONET OC-48 2.4 gigabits per second rates). Additionally, if demand for broadband services to the home and LAN/WAN connectivity through the public-switched network grows as some experts believe, then local telephone exchange carriers may install ATM switches and cross-connects for metropolitan area network (MAN) applications having aggregate bandwidths ranging from 100 Gigabits per second to 775 Gigabits per second.

By necessity, most of the current architectural research and hardware/software development for ATM switches has concentrated on switches with much smaller aggregate bandwidths which meet the more near-term needs of the marketplace. For example, most projections within the LAN/WAN community require aggregate bandwidths ranging from 150 Mbps to 12 Gigabits per second, and most of the published proposals within the telecommunications industry call for aggregate bandwidths ranging from 20 Gigabits per second to 160 Gigabits per second. Most of these ATM packet switching architectures do not scale up to larger sizes. Scaling up present ATM switching architectures produce systems that are cost prohibitive, size prohibitive, and/or physically unrealizable because of limits of the underlying technology.

For example, very common designs for large, high-throughput switches use a multi-stage interconnection network containing multiple stages of switching nodes (node-stages) interconnected by stages of links (link-stages) to provide multiple paths between input ports and output ports. Clos, Banyan and Benes networks are examples of such networks. A multiple stage network design can yield networks with very high levels of performance (low blocking probabilities, low delay, high degrees of fault tolerance, etc.), and may result in low system-level costs, because network resources (nodes and links) are time-shared by the many different paths that can be set up within the network. Physically realizing a

multistage network for a large throughput ATM packet switch is, however, a problem because of the time and processing power required by routing determinations.

Any large, high-throughput ATM switching architecture must address two fundamental issues that profoundly effect the overall performance of the resulting ATM switch. The first of these issues is cell loss due to blocking within the internal links of the distribution network (also known as the switching fabric), and the second is cell loss due to contention for output ports by two or more ATM cells that pass through the switch at the same moment in time. The first issue can usually be solved by designing a network with sufficient switching fabric (nodes and links) such that multiple paths exist between input ports and output ports. As a result, if two or more ATM cells attempt to use the same shared resource (nodes or links) within the switching fabric, each of the cells can usually find a path, thereby eliminating most of the internal network blocking problem. The second issue requires some type of method and apparatus for handling cells simultaneously arriving at the same output port.

A general technique for a switch to handle cells destined for the same output port is analyzed in an article, A Growable Packet Switch Architecture, IEEE Transactions on Communications, February, 1992, by Eng et al. and in another article The Knockout Switch, ISS AT&T Technical Papers, 1987, by Yeh et al. This general design technique, as shown in Fig. 1, segments a packet switch into two distinct parts. An Nx(FN) distribution network (which provides for N input ports) and a bank of K mxn output packet modules (which provide for a total of M=Kn output ports). Given that each of the links emanating from the distribution network is required to be terminated at one of the inputs to one of the output packet modules, it can be seen that the equation FN=Km must be satisfied. In the switch shown in FIG. 1, the switching fabric is a memory-less Nx(FN) fanout switch whose function is to route an arriving ATM cell to any of the m inputs on the output packet module connected to the cell's desired output port. The output packet module is a mxn switch with buffers that are available for storing cells that must be delayed when two or more cells contend for a particular output port. If the arriving traffic is uniformly distributed across all output ports and if the buffers within the output packet modules are sufficiently large, then the ratio m:n can always be chosen large enough to force the cell loss probability within the network to be below any desired cell loss probability level. In fact, if the network size (N) is large and if R represents the switch loading, then the cell loss probability of a network with mxn output packet modules as shown by Eng et al is given by:

$$P(\text{cell loss}) = [1-m/(nR)]\,[1-\sum_{k=0}^{m}\{(nR)^k e(-nR)\}/k!]+(nR)^m e(-nR)/m!.$$

Existing small packet switches have acceptable cell loss probabilities of approximately 10^{-12}, so any loss probability smaller than existing units are considered acceptable.

Besides the ATM cell losses because of internal contentions, in an ATM packet switch where all of the N cells arrive simultaneously at the inputs of the distribution network, the cells must be processed at each stage and path hunts must be processed before the next group of N cells arrives at the network input ports. If for example, the incoming transmission lines support SONET OC-48 2.5 Gigabits per second bit-rates, then the group of N ATM cells that arrive together must be processed and sent on to the next stage of the pipeline every 176 nano seconds (the duration of an ATM cell on a 2.5 Gigabits per second link). For large values of N, a substantial amount of processing power is therefore required to complete the path hunt operations for all N cells. (For an N=256, then at least 1.45×10^9 path hunts must be completed every second, which corresponds to an average processing rate of one path hunt every 684 pico seconds). Present commercial microprocessors can process approximately 100 million instructions per second. If each path hunt took only one instruction, these 1.45×10^9 path hunts would required the total processing power of at least 15 such microprocessors. Thus, a path hunt controller based on something other than a single commercial microprocessor will be necessary for a large ATM packet switch.

Two approaches to solving the path hunting problem can be envisioned. One approach uses in-band, i.e. self-routing, control techniques to perform the required path hunts. For in-band control techniques, the connection requests are prepended to the ATM data cells and routed through the switch along the same paths used by the following ATM payload. This approach typically requires processing elements to be distributed throughout all of the nodes in the network. This results in relatively complicated hardware within each node of the network in order to perform localized path hunting operations (on only the cells that pass through that node) when determining how to route the arriving connection requests and ATM cells. The second approach uses out-of-band control techniques whereby the controller and switch fabric are logically separated with this architecture. Connection requests are routed to the path hunting controller before the control signals resulting from the path hunt are injected into the switch fabric to set the paths. This second approach requires that the out-of-band controller have tremendous processing power, (as mentioned above), because of the many path hunt operations that must be performed in a very short period of time.

Since the path hunt operations in switches that use in-band control techniques are only based on localized traffic information and not on global information with respect to all of the switch traffic, the connections resulting from in-band

path hunting may not always be routed in optimal fashion. As a result, systems using in-band control techniques often require more switch fabric (stages and nodes) to provide the same operating characteristics as a less expensive switch based on out-of-band control techniques. In addition, out-of-band control ATM switch architectures share many similarities with the partitioning of many existing telecommunication switching and cross-connect products that have centralized controllers, so the development of a system based on a function architecture should yield fewer design problems than an architecture based on newer architectural approach. Thus, an out-of-band control ATM switch should benefit from lower overall hardware costs and have more standard architectural partitioning. On the other hand, the difficulties associated with performing path hunts in an out-of-band controller for N arriving ATM cells and the time required by a standard partitioned out-of-band controller to perform N path hunts tends to argue in favor of a newer in-band control switch architecture. For example assuming a single path hunt requires at least one read from a busy-idle memory and one write to a busy-idle memory, N path hunts require 2N accesses to memory. If N=256, then the controller must make 512 memory accesses every 176 nano seconds, so the average memory access time must be 340 pico seconds. Since 340 pico seconds memories are not commercially available, a path hunt scheme different than the present standard architectural portioning is required for any out-of band controller.

The high probability that large ATM switches will be required coupled with the uncertainties and shortcomings of present ATM architectures demonstrate a strong need in the art for a packet switch architecture that will operate with throughputs at the terabit per second levels and yet may be built using components manufactured by existing semiconductor technologies. Further, this architecture needs to be flexible to provide a telecommunications network that can evolve as customers change from non-ATM based communications to ATM based communications.

Summary of the Invention

Briefly stated, in accordance with one aspect of the invention, an advance over the prior art is achieved by providing an ATM packet switch architecture having line cards with one ATM cell period storage, a single stage, reduced interconnects switch fabric, an out-of-band controller, and output packet modules.

In accordance with an other aspect of the invention, an advance over the prior art is achieved by providing a packet switch for a switching telecommunication packet from any of a number of input lines to any of a number of output lines. This packet switch includes a number of input interfaces, each having an input port connected to a respective input line of the number of input lines and an output port. Each of the output ports is fanned out to F input ports, where F of a switching network, which has I input ports and P output ports. F is the fan-out number, which is an integer greater than 1, I is a number that is an integer multiple of the number of input lines, and P is a number that is an integer multiple of the number I of the input ports. The switching network is partitioned into a number, C, of pipes, where C is an integer having a value equal to P divided by I. A number output modules, which together have a number of output module inputs, are connected to P output ports of the switching network. Each of the output module inputs is connected to a respective output port of the P output ports of the switching network. The output modules together have a number plurality of outputs, each of these output module outputs is connected to a respective output line of said number of output lines. Each pipe of the C pipes has a path from each of the number of inputs lines that is connectable to a respective output line of the plurality of output lines. Also, the packet switch includes an out-of-band path hunter for hunting a path for a telecommunication packet from its respective input interface to its desired output line.

In accordance with another aspect of the invention, the packet switch can increase the likelihood of successfully hunting a path through the switching network by providing each pipe with a switching pattern from its input ports to its output ports that is independent from the switching patterns of the other pipes. Furthermore, the packet switch increase the likelihood of successfully hunting a path through the switching network by providing preference to predetermine an order for packets to be selected for connection through the packet switch.

Brief Description of the Drawing

FIG. 1 is a block diagram of a generalized growable packet switch;

FIG. 2 is a slightly re-drawn Fig. 1.

FIG. 3 is a block diagram of a growable packet switch in which the switch fabric is partitioned into L multiple pipes according to the present invention.

FIG. 4 is a block diagram, similar to Fig. 3, of a specific embodiment of the present invention having four pipes (L=4) and showing a configuration for the pipes.

FIG. 5 is a simplified block diagram of the embodiment shown in FIG. 4 which shows greater details of the controller.

FIG 6. illustrates the timing sequences of requests to the controller shown in FIG. 5.

FIG. 7 is a simplified block diagram of an embodiment of an output module.

FIG. 8 is an illustrative example of rolling and its operation in a plan view of an amusement park and its satellite parking lots.

FIG. 9 shows plots of calculated values of various ATM cell loss probabilities both with and without the assignment of preferences.

FIG. 10 is a simplified block diagram of a representative switch controller and its link controllers.

FIG. 11 is a detailed logic diagram of a link controller.

FIG. 12 is a state table for the link controller shown in FIG. 11.

FIGs. 13A-13D when joined together show the operation of a switch controller in response to a sequence of requests.

FIG. 14 illustrates the rolling of path hunting requests through a switch having four pipe controllers according to the present invention.

FIG. 15 shows a plot of cell loss probability versus the percentage of faulty links in an ATM switch system according to the present invention.

FIG. 16 is a simplified block diagram of a controller according to the present invention.

FIG. 17 is a block diagram of an embodiment of the present invention which carries both STM communications and ATM communications.

FIG. 18 is a block diagram, similar to Fig. 4, of a specific embodiment of the present invention having independent connections between each of the input interfaces and each of the four pipes (L=4) according to Galois field theory.

Detailed Description

Referring now to FIG. 2, a large, generalized switch 10 for ATM communications, is shown in block diagram form. ATM switch 10 has a number of input interfaces 12_0 - 12_{N-1}, a switch fabric 14, and buffered output modules 16_0 - 16_{V-1}. For ATM operation, input interfaces 12_0 - 12_{N-1}, are high speed digital amplifiers that serve as a matching networks and power amplifiers for fanning out information received on their inputs to multiple input ports of the switch fabric 14. Each of the input interfaces 12_0 - 12_{N-1} also needs a capability to store one ATM cell, as will be explained below. Similarly for ATM operation, buffered output modules 16_0-16_{V-1} are concentrators that are buffered to reduce packet loss when two or more packets are directed to and contend for the same output of outputs Out_0-Out_{N-1}.

Switch fabric 14 includes a fanout F where each of the outputs from the input interfaces 12_0-12_{N-1} is fanned out to F inputs within switch fabric 14, such that if ATM switch 10 is an NxN switch then switch fabric 14 will have FN internal inputs and FN outputs to output modules 16_0-16_{V-1}. Output Modules 16_0-16_{V-1} have a fanin or concentration factor of F in order to convert the FN outputs of the switch fabric 14 to N output module outputs Out_0-Out_{N-1}. Each output module 16_0-16_{V-1} stores arriving ATM packets in FIFO queues, and then routes the ATM packets at the front of each of these FIFO queues to their desired outputs Out_0-Out_{N-1} when the output ports are available.

Switch fabric 14 is a general distribution network which may be a network of switches, specifically crossbar switches, to provide multiple paths from each of its input ports 17_0-17_{N-1} to each of its output ports 19_0-19_{FN-1}. However, it becomes highly impractical to make an NxN switch out of a single crossbar to operate as the switching component of switch fabric 14 when the size of N exceeds 32. Thus, some other way is needed to realize the general architecture shown in FIG. 2.

Referring now to FIG. 3, an ATM switch 10A that is both practical and possible for N inputs where the size of N is at least 256, is shown. Multiple paths from each input 17_0-17_{N-1} through the switch fabric 14A are provided to prevent blocking. These multiple paths are partitioned into groups called pipes with each pipe providing exactly one path between each input port 17_0-17_{N-1} and each output port 19_0-19_{FN-1} within the network. Thus, switch fabric 14A is made up of multiple pipes 18_0-18_{L-1}. The output modules 16_0-16_{V-1} are essentially the same as the output modules shown in FIG. 2.

Switch fabric 14A, is a single stage, memoryless, and non-self routing network. Since the switch fabric 14A is not unconditionally non-blocking as a full NxN crossbar switch would be, a controller 20 is included to hunt for a path through the four pipes for each ATM cell. Since each of the pipes 18_0-18_3 contains a path that could transport the ATM cell, the real purpose of the controller 20 is to find a path that is not blocked.

For ATM switch 10A, if the number of input lines, N is equal to 256 and if each input line is operated at a standard 2.5 Gigabits per second data rate, its aggregate throughput will be 0.640 terabits per second. Scaling or growing such an ATM switch by a factor of two to 512 input lines and output lines should be straightforward and result in aggregate throughputs of greater than 1 Terabits per second. Scaling to an ATM switch size of 1024x1024 has been calculated to be within the present technology, and the architecture of the present invention is believed to be extensible even further as the speed of commercially available components increases and as new, faster technologies are developed.

Referring now to FIG. 4, a specific embodiment of an ATM switch 10A is shown. In this specific embodiment ATM switch 10A has two hundred fifty six input interfaces 12_0-12_{255} which are connected to two hundred fifty-six ATM input lines In_0-In_{255}. The outputs of the input interfaces are connected to the input ports 17_0-17_{N-1} of the switch fabric 14A. The switch fabric 14A contains a total of sixty-four 16x16 crossbar switches 15_0-15_{63} which are partitioned into four pipes 18_0-18_3. the fanout F is equal to four which if the number of output ports = FN results in 1024 output ports 19_0-19_{1023}. The output ports 19_0-19_{1023} are respectively connected to the inputs of sixteen 64x16 output packet modules 16_0-16_{15}. The sixteen 64x16 output packet modules are connected to two hundred fifty six outputs Out_0-Out_{255}. Those skilled in the art will recognize that other combinations of components could have been used, for example thirty two 32x8 output modules could have been used instead of the 64x16 output modules shown in FIG. 4.

ATM switch 10A also has a controller 20 which has the tasks of hunting and finding an available pipe through the switch fabric 14A for each ATM packet. The controller 20 uses the fact that the switch fabric 14A is partitioned into four pipes to break the pipe hunting tasks into four parallel pipe hunting tasks that are each temporally shifted by an acceptable amount. Details of one embodiment of such a controller 20 are shown in FIG. 5.

For the 0.640 Terabits per second, N=256 embodiment mentioned previously and shown in FIGs. 4 and 5, the controller 20 may be contained on approximately eight printed circuit boards. Controller 20 would accept up to 256 sixteen-bit request vectors from up to 256 line input interfaces 12_0-12_{255} and perform path hunts on each of these request vectors within each 176 nanosecond. ATM cell interval to create the 1024 sixteen-bit connect vectors used to established connections within the switch fabric 14A. This requires that controller 20 operate with a processor clock rate of at least 46 Mbps. This moderate clock rate permits the logic within the controller 20 to be implemented with off-the-shelf CMOS EPLD's or similar devices, thus making the cost of the controller 20 (in large quantities) very reasonable.

The movement of request vectors from the input interfaces 12_0-12_{255} to the controller 20 and the movement of connect vectors from the controller 20 to the crossbar switches 15_0-15_{63} of the switch fabric 14A is a challenging task, because large amounts of control information must be transported every 176 nano seconds ATM cell interval. For example, in an ATM switch containing 256 input interfaces, 256 16-bit request vectors must be transported to the controller 20 every 176 nano seconds, leading to an aggregate bandwidth of 23 Gigabits per second between the input interfaces sub-system and the controller 20 sub-system. In addition, 1024 16-bit connect vectors must be transported to the switch fabric 14A every 176 nano seconds to control the crossbars switches 15_0-15_{63}. This requires an aggregate bandwidth of 93 Gigabits per second between the controller 20 sub-system and the switch fabric 14A sub-system. This 93 Gigabits per second connect vector information can be compressed into 29 Gigabits per second (given that only one input can be routed to an output during each ATM cell interval) by standard compression techniques. However, since this control information should be delivered with high reliability, all of the control connections or control links between these sub-systems should be dually redundant (not shown in FIG 4), so there is actually 46 Gigabits per second of data moving between the input interfaces cards and the controller 20 and 58 Gigabits per second of data moving between the controller 20 and the switch fabric 14A. Preferably, high-speed serial links 22 will be used to transmit this control information. For such a case, input interfaces 12_0-12_{255} would be grouped by fours such that only sixty-four serial links would be required to move request vectors from the input interfaces 12_0-12_{255} to the controller 20, and 128 serial links would be required to move the resulting connect vectors from the controller 20 to the pipes 18_0-18_3 (assuming the aforementioned data compression techniques are applied to the connect vectors).

While the use of out-of-band control techniques does require the additional hardware cost of these high-speed serial control links 22, these links 22 cause very little increase the overall system hardware cost. Considering that the 256-input ATM switch 10A of FIGs. 4 and 5 already has 1024 high-speed serial links required to route ATM cells between the input interfaces 120_0-12_{255}, and the switch fabric 14A (when the fanout of four is included) and 1024 more high-speed serial links are used to route ATM cells from the switch fabric outputs 19_0-19_{1023} to the output packet modules 16_0-16_{15}. Thus, the addition of the 192 serial links 22 for routing of the control information increases the total number of high-speed serial links within the system by merely nine percent.

Applying the calculations of Yeh et al. from the article "The Knockout Switch" the ATM cell loss probability of the ATM switch 10A shown in FIGs. 4 and 5 is 4.34×10^{-3}, assuming that the connections of the inputs are symmetrical and not independent as will be described later in this application. This cell loss probability falls short of the acceptable ATM cell loss probability of less than 1×10^{-12} mentioned previously.

To reduce the ATM cell loss probabilities, controller 20 applies a temporal spreading technique known as rolling, which provides many statistical advantages. Rolling involves and fulfills three fundamental goals that are aimed at providing more evenly distributed traffic loads. These goals are: (1) spatially distribute the traffic evenly across all pipes 18_0-18_3 so that one pipe will only carry its proportional fraction of the traffic load, (2) spatially distribute the traffic evenly across all of the 16x16 crossbar switches 15_0-15_{63} within each pipe 18_0-18_3 so that each of the crossbar switches is equally loaded, and (3) temporally distribute the traffic that arrives in a given ATM cell period across two ATM cell periods so that the traffic load can be effectively decreased in an occasional ATM cell period when an unusually high volume of traffic exists and is destined for a particular output packet module. This effective lowering of the traffic load is accomplished by delaying some of the ATM cells arriving during a congested ATM cell interval. The cells are delayed until the next consecutive ATM cell interval when the traffic load competing for the popular resources, i.e. connections to popular output packet modules, will most likely be lower, so the delayed cells should have a higher probability of being routed in the next ATM cell interval. Since the switch fabric 14A is memoryless, the ATM cells that must wait for the next ATM cell interval are stored in their respective input interfaces 12_0-12_{255}.

In addition to satisfying these three fundamental goals of packet traffic control to distribute the load, rolling also satisfies two further very important ATM system goals. First, goal (4) is that the ATM switch 10A must guarantee that ATM cell ordering can be simply maintained when an ATM stream is re-constructed at an output packet module 16_0-16_{15} even if rolling causes some of the ATM cells within the stream to be delayed differently than others. Secondly, goal (5) is that rolling must also guarantee that the controller 20 will attempt to route every ATM cell through each of the four paths to its desired output packet module, but each of the successive path hunt attempts must occur in a more lightly-

loaded 16x16 crossbar switch so that the first attempt occurs in a 16x16 crossbar switch with many previously-routed ATM cells (and very few available paths to output packet modules) while the fourth and final path hunt attempt occurs in an 16x16 crossbar switch that is virtually empty (thereby providing many available paths to output packet modules). The rolling technique is similar to spatial path hunt techniques that pack as many calls as possible in one portion of a spatial network, which by forcing near 100% occupancy in parts of a system results in the remainder of the calls having a very high probability of being successfully routed through the remainder of the system if usage is below 100%. Thus, rolling in its fourth and final path hunt attempt provides a very high probability of an ATM cell successfully being routed. Goal (5), by packing many ATM cells in one portion of the network, superficially seems to conflict with goal (1) that requires the traffic be spatially distributed across the network. However, as will be explained below, temporal spreading provided by the rolling technique permits the network to simultaneously satisfy both goals (1) and (5).

Assuming that each of the 256 input ports 17_0-17_{N-1} of FIG. 4 has an ATM cell that needs to be routed through the distribution network, and assuming that the switch fabric 14A is composed of four pipes 18_0-18_3, then the out-of-band controller 20 may be required to perform 256x4=1024 unique path hunts for the ATM cells before the cells can be routed. To distribute the ATM cells evenly across all four pipes, the 256 ATM cells requesting connections, the rolling technique divides the requests into four groups of equal size. The first group will have path hunts performed for its ATM cells in pipe 18_0 first, then in pipe 18_1, then in pipe 18_2, and finally in pipe 18_3. The second group will have path hunts performed for its ATM cells in pipe 18_1 first, then in pipe 18_2, then in pipe 18_3, and finally in pipe 18_0. The third group will have path hunts performed for its ATM cells in pipe 18_2 first, then in pipe 18_3, then in pipe 18_0, and finally in pipe 18_1. The fourth group will have path hunts performed for its ATM cells in pipe 18_3 first, then in pipe 18_0, then in pipe 18_1, and finally in pipe 18_2. This ring-like ordering of the path hunts guarantees that the routed ATM cells are distributed evenly across all four pipes. In addition, if the ATM cells within each of the four equally sized groups are selected such that the ATM cells within a single group can be routed into exactly four of the 16 inputs on any 16x16 crossbar switch, then the routed ATM cells will also be evenly distributed across all of the 16x16 crossbar switches.

Referring now to FIGs. 5 and 6, a timing diagram for a rolling technique according to the present invention is described. To satisfy goals (1), (2), and (5) simultaneously, the out-of-band controller 20 uses the time delay/time distribution described in goal (3), and these ATM cell delays required by goal (3) must be provided during each ATM cell interval. In all cases, when a group of ATM cells is passed around the ring-like structure of controller 20 from pipe 18_3 to pipe 18_0, the controller 20 re-assigns the cells to the next ATM cell interval (period) which requires that the ATM cells be delayed by one cell period. Because of this re-assignment and delay, each cell group encounters a very lightly-loaded set of 16x16 crossbar switches for its fourth and final path hunt. An additional advantage of this rolling technique using re-assignment and delay of ATM cell intervals is that it also allows more than 64 simultaneously arriving ATM cells to be routed through the switch fabric 14A to any single output packet module 16_0-16_{15} (even though there are only 64 connections or links from the switch fabric 14A to each output packet module 16_0-16_{15}). This occurs with the rolling technique because all of the ATM cells do not need to be routed during the same ATM cell interval. Thus, the rolling technique when used in the out-of-band controller 20 results in extremely low cell loss probabilities both within the switch fabric 14A and the output modules 16_0-16_{15}, even during a transient cell interval that has an extraordinarily high traffic load.

The one ATM cell period delays incurred by some of the ATM cells as they are routed through the switch fabric 14A would normally lead to the conclusion that there would be difficulties in satisfying goal(4) of maintaining proper cell ordering. However, the ring-like ordering of the path hunts within the out-of-band controller 20 guarantees that delayed cells in a stream of ATM cells will always be routed through lower-numbered pipes than non-delayed cells (where pipe 18_0 is the lowest-numbered pipe and pipe 18_3 is the highest-numbered pipe). This information, coupled with the fact that ATM cells are delayed by at most one cell period, ensures that proper cell ordering will be maintained if the cells are extracted from the switch fabric 14A and loaded into first-in-first-out queues 174_0-174_{63} (shown in FIG. 7) of each output module of the output modules 16_0-16_{15} in the order of the lowest numbered pipe to the highest numbered pipe: pipe 18_0, pipe 18_1, pipe 18_2, and pipe 18_3.

Referring now to FIG. 7, the output module 16_0 (and the fifteen other output modules 16_1-16_{15}) may be a 64x16 embodiment of the concentrator described in U.S. Patent No. 5,412,646, entitled "ASYNCHRONOUS TRANSFER MODE SWITCH ARCHITECTURE", issued May 2, 1995, by Cyr et al. and commonly assigned to the assignee of the present invention. This U.S. patent is hereby incorporated by reference. The output module 16_0 in FIG. 7 is a specific case of the generalized concentrator shown in FIG. 4 of the above-referenced patent of Cyr et al. Since the output modules 16_0-16_{15} are well described in the above referenced application, in the interest of brevity they will not be further described here.

To provide a better understanding the equation of the rolling technique, a real-life analogy will be described with respect to FIG. 8, which is a plan view of an amusement park system 500. Consider the problem of transporting a large number of people from amusement park parking lots 511, 512, 513, or 514 to the amusement park 520 using trams to shuttle the people between the two points. Tram system 530 is composed of four tram shuttle trains each with a predetermined route, which is analogous to the four pipes of switch fabric 14A. Each tram shuttle train contains sixteen cars (representing the 16x16 crossbar switches within a particular pipe), and each shuttle car is equipped with sixteen seats (representing the output links emanating from a single 16x16 crossbar switch). In this analogy, each customer (repre-

senting an ATM cell) arrives in one of four parking lots 511. 512, 513, or 514 surrounding the amusement park 520. As a result, each customer is instantly placed in one of four groups, and since the parking lots 511-514 are the same size, each group contains an equal number of customers on the average. The customers in any single parking lot 511, 512, 513, or 514 must then divide up and stand in one of sixteen lines, where each line is associated with a respective car of the tram shuttle train. The amusement park 520 is sub-divided into sixteen different theme areas (The Past Land, The Future Land, etc.), and each of the sixteen seats of a particular tram car is labeled with the theme area to which that seat's occupant will be given admission. Before arriving in the parking lot, each customer must randomly chose one of the sixteen theme areas (representing the sixteen output packet modules 16_0-16_{15}) where he or she wishes to spend the day. Customers must then find an available seat associated with their desired theme area on one of the four trams that passes by the loading area 531, 532, 533, or 534 of their parking lot. If a customer has not found an available seat after four trams have passed by, then he or she is not permitted to enter the amusement park during that day (This harsh condition represents the loss of an ATM cell due to blocking in all four pipes of the distribution network, a small but finite possibility).

The first tram that stops at the loading area that the customer can try has already visited three other parking lot loading areas, so the customer's pre-specified seat may be full. However, if the customer does find his or her seat to be vacant on that tram, then the tram will deliver him or her straight to the amusement park 520. If the customer fails to get on the first tram, he or she must wait and try the second tram which has already visited two other parking lot loading areas. If the customer is successful at finding his or her pre-specified seat on the second tram, that tram will deliver the customer to the amusement park 520 after one more parking lot stop. If the customer fails to get on the first tram and the second tram, then he or she must wait and try the third tram which has only visited one other parking lot loading area. If the customer is successful at finding his or her seat on the third tram, that tram will deliver him or her to the amusement park 520 after two additional parking lot stops. If the customer fails to get on any of the first three trams, then the customer must wait and try the fourth and final tram. Fortunately, this tram has not visited any parking lots yet, so the arriving tram is empty, and the customer's seat will be taken only if another customer in his/her parking lot line is also trying for the same seat. The system 530 satisfies goal (5), because each of the successively arriving trams is more lightly-loaded than the previous one. Thus, a controller 20 rolling ATM cells indeed can fulfill goals (1). (2), and (5).

The rolling technique if used by itself improves the ATM cell loss probability of ATM switch 10A from $4.34\text{x}10^{-3}$ to approximately 10^{-11}. Using the analysis techniques of the article "A Growable Packet Switch Architecture" the cell loss probabilities for an ATM switch 10A that has independent connections to inputs of the switch fabric 14 according to Galois field theory and also has an out-of-band controller 20 that incorporates rolling techniques can be analytically modeled and calculated. Each of the 16x16 crossbar switches in pipe 18_0 receives an offered traffic load equal to $Ra=R_L/4+Rres$, where Rres is defined to be the fraction of the 16 inputs to a 16x16 crossbar switch that are blocked in pipe 18_3 and routed to pipe 18_0 for a re-attempt. For a first attempt at solving for the cell loss probability, let us assume that $Rres=R_L/16$. Thus, the cell loss probability of a single 16x16 crossbar switch in pipe 18_0 can be determined using the equation of Eng et al.

$$P(\text{cell loss})=[1-m/(nR_L)]\,[1-\sum_{k=0}^{m}\{(nR_L)^k e^{(-nR_L)}\}/k!]+(nR_L)^m e^{(-nR_L)}/m!;$$

where m=1, n=1, and the switch loading is given by $Ra=R_L/4+R_L/16$. Using these assignments, the resulting cell loss probability for a fully-loaded (R_L=1.0) pipe 18_0 16x16 crossbar switch can be calculated to be:

$$P(\text{cell loss in pipe } 18_0)=1.3\text{x}10^{-1}.$$

Thus, the fraction of the 16 inputs to a 16x16 crossbar that are passed to the second pipe after the first attempt is given by:

$$f1\text{-}2=Ra \times P(\text{cell loss in pipe } 18_0) = (3.13\text{x}10^{-1})(1.3\text{x}10^{-1}) = 4.06\text{x}10^{-2}.$$

By symmetry, this should have also been the same as the fraction of inputs that are passed from pipe 18_3 to pipe 18_0, so the residue assumption of $R_L/16$=0.062 above was incorrect. By refining this assumption and performing a second attempt, and now assuming that $Rres=R_L/32$. Thus, the cell loss probability of a single 16x16 crossbar switch in pipe 18_0 can be determined again using the equation of Eng et al., where m=1, n=1, and the switch loading is given by $Ra=R_L/4+R_L/32$. Using these assignments, the resulting cell loss probability for a fully-loaded (R_L=1.0) pipe 18_0 16x16 crossbar switch is calculated to be:

$$P(\text{cell loss in pipe } 18_0)=1.2\text{x}10^{-1}.$$

Thus, the fraction of the 16 inputs to a 16x16 crossbar that are passed to the second pipe after the first attempt is given by:

$$f1\text{-}2 = Ra \times P(\text{cell loss in pipe } 18_0) = (2.81\times10^{-1})(1.2\times10^{-1}) = 3.37\times10^{-2}.$$

This calculation result is very close to the assumed value of $Pres=R_L/32=3.13\times10^{-2}$, so the assumption is considered to be satisfactory. The blocked cells are sent to pipe 18_1 for subsequent path hunting, and they encounter a negligible number of ATM cells from previous attempts. Thus, the 16x16 crossbar switch in pipe 18_1 can be modeled for analysis as a growable packet switch, with m=1, n=1, and Ra = f1-2 , and the resulting cell loss probability of this model is 1.4×10^{-21}. The fraction of the 16 inputs to the 16x16 crossbar in pipe 18_1 that are passed to the pipe 18_2 is 4.2×10^{-4}. Similar arguments can be used to show that the resulting cell loss probability for cells entering pipe 18_2 is 1.9×10^{-4}, and the resulting fraction of the 16 inputs to a 16x16 crossbar passed to pipe 18_3 is 7.9×10^{-8}. The resulting ATM cell loss probability in pipe 18_3 is 3.7×10^{8}, and the fraction of the 16 inputs to a 16x16 crossbar not routed in pipe 18_3 (and therefore not routed in all four pipe attempts) is 2.9×10^{-15}. Thus, through the use of the rolling techniques within the out-of-band controller 20, the ATM cell loss probability of an ATM switch 10A with independent connections at the inputs of its switch fabric 14A can be decreased from an unacceptable value of 1.47×10^{-6} to an acceptable value of 2.9×10^{-15}.

A preference technique may be used in conjunction with the rolling technique described above to decrease the cell loss probability of an ATM switch 10A even further. Referring back to FIG. 8 and the amusement park analogy, some form of arbitration was required at the tram loading areas to determine which of the customers in the line will be given a particular seat on the tram when more than one customer is requesting the same seat. Similarly, the out-of-band controller 20 must provide an arbitration scheme for selecting which of the arriving ATM cells will be assigned a particular link whenever two or more cells request access to the same link. The arbitration scheme used can have an advantageous effect on the ATM cell loss probabilities.

One possible arbitration scheme is a random scheme to determine which of the ATM cells is assigned the link. The random selection scheme is the scheme assumed for the analysis of the rolling technique presented above. However, other arbitration schemes are possible, and one particular arbitration scheme that has advantageous results is called the preference scheme. The preference arbitration scheme assigns a preference weight to each of the ATM cells in a particular grouping. ATM cells with higher preference weights are given precedence over ATM cells with lower preference weights whenever two or more cells request access to the same link. As a result, an effective hierarchy is created within the groupings of ATM cells.

The creation of a hierarchy may superficially seem to produce undesirable characteristics within the switch fabric 14A, because customers with high preference weights will be offered better service than customers with low preference weights. In fact, the one customer with the highest preference weight within each group can never have his or her ATM cell blocked by another customer's ATM cell. Although this may seem unfair, a detailed analysis of the effects of imposing this hierarchy indicates that it actually leads to improved performance, i.e. lower cell loss probabilities, for all customers-even for the customer at the bottom of the hierarchy with the very lowest preference weight.

The results of this analysis are summarized in FIG. 9, where the probability of loss of an ATM cell ; i.e., the probability of a cell not being assigned to an available path, is shown as a function of the number of path hunts that were attempted in different pipes by the out-of-band controller 20. In this analysis, it was assumed that the group sizes were four- i.e., up to four ATM cells could simultaneously compete for access to the same link. As a result, four different preference weights were assigned to create a hierarchy for the four input ports associated with each group. The preference weight associated with a particular input port is assumed to be a fixed constant that does not vary with time. The resulting plots 901, 902, 903 and 904 in FIG. 9 indicate that the cell loss probability decreases as more path hunts in more pipes are performed, but it also shows that the inputs with the lower preference weights 903, 904 have higher cell loss probabilities than the inputs with higher preference weights 901, 902, as might be expected. Super-imposed on these plots is a similar plot 910 which indicates the probability of not being served when a random selection arbitration scheme is used instead of the hierarchy arbitration scheme. The surprising and unexpected results are that after path hunt attempts in four different pipes, the random selection arbitration scheme produces cell loss probabilities which are higher than the average of the cell loss probabilities for the hierarchy arbitration scheme. In fact, the plot 910 of the random selection arbitration scheme shows an average cell loss probabilities for all of the input ports which are notably higher than the plots 903 and 904 which are the average cell loss probabilities for even the input ports with the lowest preference weights within the hierarchy arbitration scheme. This phenomenon can be explained by the fact that after three sets of path hunts in three different pipes, the distribution of ATM cell requests entering the fourth pipe is very different depending on whether the random or preferences arbitration scheme is used. In the random selection arbitration scheme, there is a small but equal probability that all of the ATM cells are requesting a path. However, in the hierarchy arbitration scheme, most of the ATM cells with higher preference weights will be requesting a path with a probability of practically zero, while the ATM cell with the lowest preference weight will be requesting a path with a sizable probability, because that particular ATM cell may have been denied access to links in all three of its previous path hunt attempts. However, a single request arriving with a high probability at the fourth and last path hunter in the controller will lead to more routed ATM cells than

many requests arriving with low probability, because the single request can always be satisfied since contention for an output link will never occur.

As a result, it seems apparent from the plots in FIG. 9 that by assigning preference weights to the input ports and by using a hierarchy arbitration method to resolve link contention and route paths in the out-of-band controller, the worst-case cell loss probability of the switch fabric 14A can be decreased from 2.9×10^{-15} that was achieved by the introduction of the rolling technique to an even lower value of 2.4×10^{-16}. It is worth noting that input ports that are assigned higher preference weights will encounter even lower cell loss probabilities as indicated in FIG. 9.

Referring back to FIG. 5, in order to provide a physical embodiment of the rolling and preference methods, the ATM switch 10A is segmented in to four basic sub-systems. These four sub-groups consist of the input interfaces 12_0-12_{255}, the output modules 16_0-16_{15}, the switch fabric 14A, and the out-of-band controller 20.

The input interfaces 12_0-12_{255} within the network provide the necessary interfaces between the incoming transmission links and the links connected to the switch fabric 14A and the out-of-band controller 20. As a result, the input interfaces 12_0-12_{255} must provide a termination for the input transmission line. For example, if the input transmission line is a SONET link, then the input interface must provide for clock recovery, link error detection, SONET pointer processing and frame delineation, ATM cell extraction, and an elastic storage function to synchronize the arriving ATM cells to the system clock within the distribution network. The extracted ATM cells are then loaded into a FIFO buffer of the input interface. The input interface must also read ATM cells from the FIFO buffer and extract the ATM header from the cell. The VPI/VCI field of each ATM header is then used as an address into a translation table located on the input interface. The output of the translation table provides a new VPI/VCI field and the address of the output packet module to which the ATM cell is to be routed. The new VPI/VCI field is written into the ATM cell as a replacement for the old VPI/VCI field, while the output module address is routed as a request vector to the out-of-band controller 20 for the controller fabric 14A. Since the amount of processing time required by the out-of-band controller 20 is a fixed value, the input interface simply holds the ATM cell in a buffer until the out-of-band controller 20 has completed its path hunt and has relayed the results into the switch fabric 14A. Once the switch fabric 14A is loaded with the new switch settings to appropriately route the ATM cell, the input interface can inject the ATM cell into the switch fabric 14A and it will be automatically routed through the switch fabric 14A to its desired output module 16_0-16_{15}. It should be noted that each input interface 12_0-12_{255} actually is provided with one link to each of the four pipes 18_0-18_3 of the switch fabric 14A. In addition, the use of rolling (i.e. temporal spreading) within the switch fabric 14A may require a copy of the ATM cell to be injected into each of the four links during any one of two consecutive ATM cell intervals. As a result, the timing within the input interfaces 12_0-12_{255} must be tightly coupled and synchronized to the timing of the rest of the sub-systems within the ATM switch 10A.

Each of the two hundred fifty six input interfaces 12_0-12_{255} in FIG. 5 are numbered with an address ranging from 0 to 255, but each input interface is also assigned an alias address given by a letter between A and P. These alias addresses are used to identify which input port the input interfaces will connect to within the switch fabric 14A. The actual set of four crossbar switches to which a particular input interface is connected is determined by Galois field techniques. These techniques guarantee independence between all of the inputs on any 16x16 crossbar switch of any pipe.

Each of the sixteen output modules 16_0-16_{15} in FIG. 5 is labeled with addresses ranging from AA to PP, and each output module performs an important function within the ATM switch 10A. Each of the output modules 16_0-16_{15} within FIG. 5 provides terminations for a respective set of sixty-four links emanating from the switch fabric 14A. Each output module 16_0-16_{15} also provides two basic functions: it provides a small degree of space switching to route each ATM cell arriving on one of the sixty-four inputs to the desired one of the sixteen output ports, and it provides buffering of ATM cells to handle the problems associated with multiple packets that are simultaneously destined for the same output Out_0-Out_{255}.

There are many ways for these two functions to be implemented. The most straight-forward approach would probably construct a shared memory switch that could perform sixty-four memory writes and sixteen memory reads within an ATM cell interval (176 nano seconds). The memory could then be treated as sixteen disjoint linked lists (one for each output Out_0-Out_{255}) along with a seventeenth linked list containing idle memory locations. Although simple, this approach requires eighty memory accesses every 176 nano seconds, so it would demand memories with 2.2 nano seconds access times. An alternate approach would split each 64x16 output module 16_0-16_{15} into a 64x16 concentrator and a 16x16 shared memory switch. The concentrator would be a memory system that provides for sixty-four writes and sixteen reads every ATM cell interval, but the memory size could be small (and memory speeds could be fast) since the buffering required for output contention problems is not provided in this memory. In addition, the 64x16 concentrator could be implemented as a single linked list spread out across sixty-four distinct memory chips. As a result, each memory chip would require only one write and up to sixteen reads for every ATM cell interval. The 16x16 shared memory switch only performs thirty-two memory accesses every ATM cell interval, so slower (and larger) memories could be used, and the buffering for output contention problems could be provided in this shared memory portion of the output module. Thus, this latter arrangement is the more practical alternative for an output module.

The switch fabric 14A is essentially a group of small circuit switches that provide the required connectivity between the input interfaces and the output modules in response to the control signals generated by the out-of-band controller 20. In the embodiment of the ATM switch 10A shown in FIG. 5, the switch fabric 14A is composed of sixty-four 16x16

crossbar switches, where disjoint groups of sixteen switches comprise a pipe. The four pipes are labeled pipe 18_0, pipe 18_1, pipe 18_2, and pipe 18_3, and the sixteen 16x16 crossbar switches within a given pipe are labeled switch 0-15. The crossbar switches must be capable of receiving the control signals generated by the out-of-band controller 20 and must re configure all of the switch settings during a guard-band interval between consecutive ATM cells. Each 16x16 crossbar switch supports sixteen inputs labeled input A through input P, and each 16x16 crossbar switch also supports sixteen outputs labeled output AA to output PP. It was noted above that each input interface connects to a different 16x16 crossbar in each of the four pipes 18_0-18_3, but it should now be noted that an input interface that connects to input X in pipe 18_0 is required to be connected to input X in the other three pipes 18_1-18_3 as well, where X is an element of the set {A,B,...,P}. The actual connections between the input interfaces 12_0-12_{255} and the crossbar switches within the switch fabric 14A are determined using Galois field theory techniques that were referenced above. These techniques guarantee independence between input ports for routing within switches in each pipe of the switch fabric 14A. FIG. 5 also illustrates that output YY from each of the sixty-four crossbar switches is routed to one of the sixty-four inputs on the 64x16 output module labeled YY, where YY is an element of the set {AA,BB,..,PP}.

The basic function of the out-of-band controller 20 for the switch fabric 14A is to determine through which of the four pipes 18_0-18_3 a particular ATM cell may be routed. Once the out-of-band controller 20 has successfully determined a pipe through which the ATM cell is to be routed without being blocked, the task of setting up the path through the pipe is simple, because by the definition of a pipe, there will exist only one path within the pipe between the input port of the arriving ATM cell and the desired output module. As a result, the fundamental path hunting task of a switching network is essentially reduced to the simpler task of pipe hunting in the ATM switch 10A.

The out-of-band controller 20 still requires a large busy-idle table to identify the status of each of the intermediate (FN) links between the 16x16 crossbar switches of the switch fabric 14A and the output modules 16_0-16_{15} as busy and unavailable or idle and available. However, this large busy-idle table may be sub-divided into many small busy-idle tables that the controller 20 can access in parallel, and thereby perform many pipe hunting operations in parallel. There are many ways to implement the controller 20 for a large switch having the general growable packet switch architecture. In the extreme case, four levels of parallelism may be applied to the architecture of the controller 20 to perform pipe hunting. One embodiment that uses three levels of parallelism will be described in detail, first and then a fourth level of parallelism for the controller 20 will be discussed.

The first level of parallelism is obtained by providing each of the four pipes 18_0-18_3 with a respective pipe hunt controller 24_0-24_3. This level of parallelism allows pipe hunting to be carried out in all four pipe hunt controllers 24_0-24_3 simultaneously. The second level of parallelism is obtained by providing switch controllers 26_0-26_{63}, with sixteen switch controllers within each pipe hunt controller 24_0-24_3. A unique switch controller 26_0-26_{63} is respectively associated with each of the 16x16 switches within each pipe of the switch fabric 14A. As a result, pipe hunting operations can be carried out in parallel within all sixteen of the switch controllers of each pipe hunt controller 24_0-24_3. The third level of parallelism is obtained by permitting each of the switch controllers 26_0-26_{63} to perform parallel processing over all sixteen of the output links attached to its respective 16x16 crossbar switch. Effectively, each of the switch controllers 26_0-26_{63} reads sixteen busy-idle bits from its busy-idle memory in parallel, performs parallel pipe hunting operations based on those sixteen bits, and then writes the sixteen resulting busy-idle bits into its respective busy-idle memory in parallel with the other busy-idle memories. A representative switch controller 26_0 of the sixty four switch controllers 26_0-26_{63} is shown in FIG. 10. The concurrent processing of sixteen busy-idle bits is accomplished by providing switch controller 26_0 sixteen unique link controllers AA-PP, each of the link controllers AA-PP is assigned the task of processing busy-idle bits for one intermediate link between its portion of the switch fabric 14A and its respective output modules. In the embodiment shown in FIG. 10, the large busy-idle memory required to control switch 10A has been divided into many single bit memories, busy-idle flip-flops, with each single bit, busy-idle memory being logically and physically associated with its respective link controller AA-PP.

The general data flow for request vectors generated by the input interfaces 12_0-12_{255} is shown in Fig. 5. For example, input interface 12_0 in Fig. 5 routes its request vector to pipe hunting controller 24_0 where it is poked into the pipe hunting ring (i.e. controller 20), and the rolling scheme requires the request vector to be looped through pipe hunt controller 24_1, pipe hunt controller 24_2, and pipe hunt controller 24_3 as it circulates around the ring. In general, each of the input interfaces 12_0-12_{255} produces one request vector, and each request vector will contain a number of bits equal to the number of output modules within the system. The request vector from a single input interface in FIG. 5 is thus a sixteen-bit data word, where each bit of the request vector points to one of the sixteen output modules. If an ATM cell within a input interface is requesting a connection to an output port on the i-th output module, then bit i within the request vector will be set to a logic "1" and all other bits within the request vector will be set to a logic "0". When the controller 20 receives this particular request vector from the input interface, it can then identify that a path is required between the source input interface and the i-th output module.

The entire sixteen-bit request vector from a input interface is routed via a respective control connection 21_0-21_{255} to one of the four pipe hunt controllers 24_0-24_3, and the controller 20 pokes the vector into one of the sixteen switch controllers associated with that particular pipe hunt controller. As shown in FIG. 10, the sixteen bits of the request vector are injected into a switch controller and are distributed across all sixteen of the link controllers within that particular

switch controller. Each link controller is associated with a single link between the crossbar switches and the output modules, and it essentially processes one bit of the sixteen-bit request vector. This finite state machine circuitry that is associated with a single link controller consists of one flip-flop (the single-bit memory required to store the busy-idle bit associated with this link controller's link) and four logic gates. A state table description of the link controller operation is given in FIG 12, where the state variable is defined by the busy-idle bit. The link controller hardware provides for one request vector input bit, designated request-in; one request vector output bit, designated request-out; and one connection vector output bit, designated connect. The request vector input bit is a logic "1" if the input desires a connection through the link associated with this link controller-otherwise, it is a logic "0". The request vector output bit is a logic "1" if the logic "1" input request vector bit was not satisfied by this particular link controller-otherwise, it is a logic "0". The connect vector output bit is a logic "1" if the logic "1" input request vector bit was satisfied by this particular link controller indicating the ATM cell will be routed to its desired output module through the link associated with this link controller- otherwise, it is a logic "0". The busy-idle flip-flop in Fig. 10 is reset to the logic "0" (idle) state at the beginning of each ATM cell slot, so the first request vector bit that enters the link controller with a logic "1" request is assigned the link (creating a logic "1" connect vector bit and a logic "0" output request vector bit) and sets the busy-idle flip-flop to the logic "1" (busy) state. Any subsequent request vector bits that enter the link controller during this particular ATM cell slot will be denied a connection through this link (forcing a logic "0" output on the connect vector bit and creating an output request vector bit that is identical to the input request vector bit). A time-lapsed view of several consecutive sixteen-bit request vectors passing through a single switch controller is shown in Fig. 12, along with the resulting states of the busy-idle bits stored within the switch controller. The resulting output request vectors and output connect vectors illustrate the general operation of each of the pipe hunt controllers 24_0-24_3.

The use of rolling within the controller 20 requires a very precise temporal ordering of two fundamental events: poking and busy-idle flip-flop clearing. The timing diagram of FIG. 13 illustrates the synchronization and data flow that might be used for the logic within the controller 20. As indicated by the timing diagram, the flow of data around the ring of controller 20 is from pipe controller 24_0 to pipe controller 24_1 to pipe controller 24_2 to pipe controller 24_3 and back to pipe controller 24_0. Request vectors generated by input interfaces with alias addresses A, B, C, and D are poked into pipe controller 24_0. Request vectors generated by input interfaces with alias addresses E, F, G, and H are poked into pipe controller 24_1. Request vectors generated by input interfaces with alias addresses I, J, K, and L are poked into pipe controller 24_2. Request vectors generated by input interfaces with alias addresses M, N, O, and P are poked into pipe controller 24_3. The poking times and busy-idle bit clearing times take place at different moments within each of the pipe hunt controllers 24_0-24_3. From the point of view of any pipe controller, the request vector bits flow through the pipe controller in alphabetical order (A to P) if one ignores the busy-idle bit clearing times. This ordering guarantees that the aforementioned advantages of preferences will be realized within the controller 20, because the request vector generated from a input interface with alias address A will always be given precedence over the request vectors generated from input interfaces with alias addresses B, C, and D, etc.

The benefits derived from forced independence between the inputs on a particular 16x16 crossbar switch produce a slight increase in the complexity of the pipe hunter circuitry. Because of the independent connections between the input interfaces and the switch fabric 14A, which independence is assured by the use of Galois field theory, a request vector from a single input interface must be appropriately routed to several different switch controllers in each of the stages in the pipe hunting ring. The mixing nature of the Galois field theory generated connections requires each input interface 12_0-12_{255} to be connected to a different set of 16x16 crossbar switches within the switch fabric 14A, and as a consequence, it also requires request vectors generated on different input interfaces to be routed through entirely different sets of switch controllers within the controller 20. Since request vectors are time-multiplexed on links within the controller 20, all of the request vectors (within a particular ATM cell slot) that are expelled from a particular switch controller in one pipe hunter stage must (by definition) be routed to different switch controllers in the next pipe hunter stage. To provide this dynamic routing of the request vectors, each pipe hunt controller 24_0, 24_1,24_2 and 24_3 is connected to a respective small switching network 30_0. 30_1, 30_2 and 30_3, shown in FIG. 5. Alternatively, simple multiplexers may be used instead of switching networks 30_0. 30_1, 30_2 and 30_3, thereby greatly decreasing costs for the controller 20. Fortunately, the required configurations of these small switching networks 30_0, 30_1, 30_2 and 30_3,(or multiplexers) are cyclic with a period equal to the ATM cell period, and the required configurations can be determined *a priori* and can therefore be "hard-coded" into the small switching networks (multiplexers) during the design of the circuitry of the controller 20.

As mentioned previously, ATM switch 10A shown in FIG. 5 might be scaled such that the number of input lines were 512, 1024 or even higher. For those size switches, assuming that the input lines are carrying 2.5 Gigabits per second data rates, the aggregate throughput would be over 1.0 Terabits per second. For switches of that size, a fourth level of parallelism may be needed to provide sufficient processing power for the controller 20 to hunt for all the paths through all the pipes in real time. For ATM switches with 512 and 1024 input lines, the data rates on connections within their respective controllers are 204 Mbps and 386 Mbps, which is considerably higher that the 113 Mbps rate of the 256 input line version of ATM switch 10A.

The basic idea behind the fourth level of parallelism is a modification of the previously described controller 20 design which requires that request vectors be routed through the pipe hunter stages in parallel. In particular, all of the request

vectors that are poked into a particular pipe are routed through the pipe hunter stages together, and these request vectors are said to comprise a poke group. In the embodiment shown in FIG. 5, this approach to the design of controller 20 creates four poke groups of sixteen-bit request vectors, so each poke group contains sixty-four bits. The four poke groups can be labeled with a concatenation of the four alias labels on the request vectors. As a result, the four poke groups for the re-designed pipe hunter of FIG. 5 are called ABCD, EFGH, IJKL, and MNOP. It is important to note that whenever a single sixty-four bit ABCD poke group is being routed through one of the switch controllers in pipe controller a of FIG. 5, there is also a sixty-four bit ABCD poke group being routed through each of the other fifteen switch controllers in pipe controller 24_0. As a result, there are a total of 1024 request vector bits associated with sixteen ABCD poke groups that are being routed through pipe 18_0 at a single instant of time. The modified controller 20 processes the request vectors for all N input ports (by passing them through all four pipe hunt controllers 24_0-24_3) every eight clock cycles, and since this task must be completed within a single 176 nano second ATM cell interval, the required clock rate within the controller 20 is 46 megabits per second regardless of the size (aggregate throughput) of the NxN ATM switch. As a result, since the controller 20 must perform eight processing steps (regardless of the network size), the process is said to be an O(1) path hunt algorithm. During the execution of this O(1) path hunt algorithm for the N=256 input ATM switch 10A of FIG. 5, the equivalent of 16,384 link controller path hunts and 16,384 link controller path hunt checks are performed every 176 nano second, so if each path hunt is considered to be an instruction execution and each path hunt check is considered to be an instruction execution, then the controller 20 can be viewed as a parallel processor capable of sustaining a 186 giga-instructions per second processing rate. The trade-off for maintaining a reasonable data rate in the controller 20 (regardless of size) is an increase in link controller logic complexity and an increase in signal connections passing between successive stages of the controller 20 as the size is increases. ATM switch designs with aggregate throughputs in excess of 1 Terabits per second will require between 4096 and 32,768 signals at 46 megabits per second to be routed between successive stages of the controller, i.e. between pipe hunt controllers 24_0-24_3.

In addition to increasing the number of signals between pipe hunt controller stages, the use of parallelism within the controller 20 also requires a slight increase in the hardware requirements for each link controller, because each link controller must now support a parallel path hunt on four bits within the poke group. The extra hardware added to the controller 20 by the fourth level of parallelism should be offset by the resulting lower processing rate.

In addition to large throughputs and low data losses, an important and essential attribute for any switching product that will be used in the public switched network is fault tolerance, in order to provide a very high level of availability. A switching system that is fault-tolerant must display most, if not all, of the following attributes: 1) the ability to detect that a fault exists, 2) the ability to locate and identify the faulty component, 3) the ability to avoid the faulty component by detouring traffic through alternate paths within the network, 4) the ability to provide an acceptable level of performance (cell loss probability) even in the presence of a small percentage of faulty components, and 5) the ability to permit maintenance personnel to easily repair the faulty component (e.g., by swapping a board), and 6) the ability to provide an acceptable level of performance, i.e. cell loss probability, even when the faulty component is being repaired. These attributes typically require some level of redundancy within the switch paths to satisfy the requirements of attributes 3 and 4, and they also require redundancy at a next level higher within the network fabric and controller to satisfy the requirements of attributes 5 and 6.

A large benefit of using out-of-band control techniques within an ATM switch is derived from the fact that existing fault-tolerance techniques that have been applied for years in circuit switches can be re-used in the ATM switch 10A. As a result, all six of the above fault-tolerance requirements can be easily satisfied within the general architecture of ATM switch 10A. For example, FIG. 15 shows a plot of a simulation of the cell loss probability within the architecture of FIG. 5 when faulty links are added to the switch fabric 14A. It can be seen that while the cell loss probability increases as fault links are added, up to 0.5 percent of the links can be faulty before the cell loss probability exceeds the maximum acceptable level of 1×10^{-12}. This is a direct result of the fact that the architecture in FIG. 5 provides four paths between each input port and output port. Faulty paths can be rapidly identified by parity or CRC checks at the output modules 16_0-16_{15}. If an error is detected, the controller 20 already knows what path the corrupted ATM cell had been routed through, so it can check the path by sending an "interrogation ATM cell" through the switch fabric 14A. If the interrogation ATM cell is also corrupted, then the path should be taken out of service by writing the busy-idle bit for the path to the maintenance busy state, which is not cleared even when the global clear is sent at the end of each ATM cell period.

Although the present ATM standard does not define or require a service that supports variable length cells or packets, the evolutionary tendencies of the telecommunication industry are towards ATM cells with lengths that differ from the initially defined 53-byte standard. This change may evolve as a result of the satisfaction (or dissatisfaction) that different users may find as they begin to experiment with different applications being transported over the ATM packet lines and networks. For example, the very fact that the current ATM cell length represents a compromise between the voice and data communities indicates that there may be some customers in the future who are not entirely satisfied with the offered cell size. Should such customers become organized, it is possible that another cell length (other than 53 bytes) may be requested. For example, the CATV industry is already considering cell sizes greater than 53 bytes for transporting MPEG-2 digital video streams. Although larger packets might be routed inside multiple 53 byte ATM cells, the resulting bandwidth

inefficiencies may ultimately lead to the desire for a new cell length standard. Thus, it is desirable to have an ATM switch architecture that can adapt to changes in cell or packet length.

With only a few minor modifications to the input interfaces 12_0-12_{255} and the controller 20, operation of ATM switch 10A with cells of arbitrary cell lengths is provided. This is accomplished by allowing arbitrary length that are an integer multiple of the basic cell period, which can be 53 bytes or some other desired length. The ATM switch 10A may be modified so readily to support arbitrary cell lengths primarily because ATM switch 10A is essentially a circuit switch, i.e. an arbitrarily long message switch, with the update capability of a very fast out-of-band path hunt processor. For fixed cell length operation described previously, the controller 20 of the switch 10A performs path hunts for cells that arrive on potentially all of the N input ports, and it must then set up the N paths for all of these routed cells. At the end of the 176 nano second cell interval, all of these N paths are globally torn down, making all of the network connections idle for the next cell interval when the entire process is repeated. If variable length cells are permitted, then it should be apparent that the global tear-down of all N paths at the end of a cell interval is no longer permitted. In fact, all of the paths from one cell interval must be left established in the next cell interval, and individual path tear-downs must be implemented whenever the termination of a cell is identified by its respective input interface 12_0-12_{255}. Thus, to modify ATM switch 10A to handle variable length cells requires that each input interface 12_0-12_{255} be modified to be capable of identifying the start and end of each cell, either with fixed and unique start and end patterns, a unique start pattern coupled with a cell length identifier contained within the cell, or some other type of indication. Each modified input interface must then be capable of sending two different types of request vectors to a modified controller 20': one to request a path set-up and one to request a path tear-down. This can be accomplished by adding a single bit to the 16-bit request vector shown in FIGs. 13A-13D, where the additional bit is used to indicate whether the request is a set-up request or a tear-down request. Upon reception of the request vector, the controller 20' routes the request vector through all four of the pipe controllers 24_0-24_3 (as before), but the link controllers must now be capable of both setting and resetting the busy-idle flip-flops in response to the different types of request vectors. In addition, the link controller associated with a particular output link must also maintain a memory indicating which input presently has a path established to its output link, because only that input is permitted to tear down the path to the output link. The hardware required for all of these functions within a link controller is illustrated in FIG. 16. It should be noted that the inclusion of variable length cell routing within the distribution network requires that the processing rate within the controller 20' be increased by a factor of two, because it is possible that every input may require a single path set-up and a single path tear-down every ATM cell interval. It should also be noted that the inclusion of variable length cell routing does not preclude the implementation of any of the other features of the switch 10A that have been mentioned previously.

The use of variable length cells within the switch fabric 14A also requires that the output modules 16_0-16_{15} be modified to route cells with different cell lengths. The lengths of the buffers would have to be increased to accommodate at least four times the longest cell or packet that can be communicated.

The rapid acceptance of ATM within the LAN and WAN communities indicates that ATM may already be developing a strong foot-hold in the private-switched network environment. As a result, it may be only a matter of time before there is a strong demand for ATM services in the public-switched network environment. However, when or if that strong demand occurs is uncertain. There is some question within the telecommunications industry of the ability of ATM to efficiently provide services that are inherently constant-bit-rate services (voice and video), and also the ability of ATM networks to effectively route traffic in a network where the traffic is highly correlated rather than random. Because of this uncertainty, ATM service providers and ATM switch vendors must proceed cautiously. A switch based on the switch fabric 14A is a sensible system in such uncertain times because the architecture is flexible enough to provide both packet switching (ATM) and circuit switching (STM) communications at the same time. Such a switch 10B is shown in FIG. 17. Since STM switching is a form of circuit switching, and since the switch fabric 14A is essentially a circuit switch with very fast path hunting capabilities, the switch fabric 14A is well suited for the routing of STM traffic. A slightly different controller 620 is required for STM traffic, and the input interfaces 612 and output modules 616 provide time-slot interchanger functions required by circuit switching equipment, but the single stage switch fabric 14A can remain unaltered in a combination STM and ATM switch, or in a wholly STM switch. Simulations have been written to analyze the operation of the switch fabric 14A in a wholly STM environment, where an N=256 ATM switch is modified to implement an N=128 STM switch. The resulting blocking probability of this N= 128 STM switch has been calculated from this simulation to be less than $1x10^{-9}$. This is a very acceptable value in a circuit switched environment where packet loss is not an issue. Thus, switch 10B can vary the percentage of STM traffic it carries up to 100 per cent. This flexibility greatly reduces or eliminates the possible financial consequences of the uncertainty in the demands of customers for future ATM and STM services.

The switch fabric 14A is essentially technology-independent. An embodiment using free-space digital optics as the interconnection technology within the switch fabric is contemplated. The 16x16 crossbar switches within the switch fabric 14A will be implemented with FET-SEED device arrays. Such an approach may provide many benefits within the switch fabric 14A, because the resulting design based on optical interconnections may have lower levels of signal crosstalk, lower chip counts due to increased device integration, lower signal skew, and lower overall power dissipation, which results in simpler thermal management techniques within the switch fabric 14A.

The packet cell loss of the switch fabric 14A is decreased by the use of a more complicated mapping function. Referring to FIG. 18, a methodology is identified that uniquely assigns each input port to a single input on one of the 16x16 crossbar switches in each of the pipes $18B_0$-$18B_3$ of switch fabric 14B so that two input ports will connect to a common 16x16 crossbar switch at most once in the entire switch fabric, thereby guaranteeing independence of the inputs. The mathematics of finite field theory, i.e., Galois field theory, provide the desired methodology and by applying mapping functions based on Galois field theory between the input ports of the fabric 14B and the 16x16 crossbar switches 17_0-0 to 17_0-15, 17_1-0 to 17_1-15, 17_2-0 to 17_2-15 and 17_3-0 to 17_3-15 independent connections may be determined.

To determine independent connection for the switch fabric 14B, assume that input port (I) can be represented by the eight-bit binary number (i_7, i_6, i_5, i_4, i_3, i_2, i_1, i_0), and assume that the 16x16 crossbar switch S_0 (I) to which it is routed in pipe θ , e.g. pipe 0 through pipe 3 corresponding to pipe $18B_0$ through $18B_3$, can be represented by the six-bit binary number (s_5, s_4, s_3, s_2, s_1, s_0)θ . Using Galois field theory, many different sets of acceptable mapping functions that guarantee independence of input ports may be identified. One possible set of acceptable mapping functions for pipes $18B_0$ - $18B_3$ for input (i_7, i_6, i_5, i_4, i_3, i_2, i_0) is given below:

$$(s_5, s_4, s_3, s_2, s_1, s_0)\ 0=(0, 0, i_3, i_2, i_1, i_0)$$

$$(s_5, s_4, s_3, s_2, s_1, s_0)\ 1=(0, 1, i_7 \text{ XOR } i_3, i_6 \text{ XOR } i_2, i_5 \text{ XOR } i_1, i_4 \text{ XOR } i_0)$$

$$(s_5, s_4, s_3, s_2, s_1, s_0)\ 2=(1, 0, i_7 \text{ XOR } i_2, i_6 \text{ XOR } i_1, i_5 \text{ XOR } i_0 \text{ XOR } i_7, i_4 \text{ XOR } i_7)$$

$$(s_5, s_4, s_3, s_2, s_1, s_0)\ 3=(1, 1, i_7 \text{ XOR } i_1, i_6 \text{ XOR } i_0 \text{ XOR } i_7, i_5 \text{ XOR } i_7 \text{ XOR } i_2, i_4 \text{ XOR } I_2).$$

In addition, network input (i_7, i_6, i_5, i_4, i_3, i_2, i_1, i_0) is connected to inlet i_7, i_6, i_5, i_4) on each of the 16x16 crossbar switches to which it is routed. When these link mappings, which are based on Galois field theory, are employed, the resulting set of interconnections between the input ports and the 16x16 crossbar switches of the switching fabric 14B are Galois connections. The results of the above mapping function when applied to the system shown in FIG. 18, are listed in the following pages. The connections listed are between switch fabric inputs 0-255 and four input ports on the four 16x16 crossbar switches in respective pipes $18B_0$-$18B_3$. The list is organized in the following manner. Each input #0-255, which corresponds to a respective output of input interfaces 12_0-12_{255}, is fanned out to four switch input ports all having the same switch input port number. However, these four switch input ports (which all have the same number) are each on a different numbered 16x16 crossbar switch for each respective pipe. For example, input #21 connects to input port 5 of 16x16 crossbar switch #1 of pipe $18B_0$, input port 5 of 16x16 crossbar switch #4 of pipe $18B_1$, input port 5 of 16x16 crossbar switch #11 of pipe $18B_2$ and input port 5 of 16x16 crossbar switch #6. Thus, all of the independent connections according to the embodiment of the present invention shown in FIG. 5 are listed in the following pages.

input #: 0
switch input port #: 0
pipe 0 switch #: 0
pipe 1 switch #: 0
pipe 2 switch #: 0
pipe 3 switch #: 0

input #: 1
switch input port #: 1
pipe 0 switch #: 0
pipe 1 switch #: 1
pipe 2 switch #: 2
pipe 3 switch #: 4

input #: 2
switch input port #: 2
pipe 0 switch #: 0
pipe 1 switch #: 2
pipe 2 switch #: 4
pipe 3 switch #: 8

input #: 3
switch input port #: 3
pipe 0 switch #: 0
pipe 1 switch #: 3
pipe 2 switch #: 6
pipe 3 switch #: 12

input #: 4
switch input port #: 4
pipe 0 switch #: 0
pipe 1 switch #: 4
pipe 2 switch #: 8
pipe 3 switch #: 3

input #: 5
switch input port #: 5
pipe 0 switch #: 0
pipe 1 switch #: 5
pipe 2 switch #: 10
pipe 3 switch #: 7

input #: 128
switch input port #: 0
pipe 0 switch #: 8
pipe 1 switch #: 8
pipe 2 switch #: 8
pipe 3 switch #: 8

input #: 129
switch input port #: 1
pipe 0 switch #: 8
pipe 1 switch #: 9
pipe 2 switch #: 10
pipe 3 switch #: 12

input #: 130
switch input port #: 2
pipe 0 switch #: 8
pipe 1 switch #: 10
pipe 2 switch #: 12
pipe 3 switch #: 0

input #: 131
switch input port #: 3
pipe 0 switch #: 8
pipe 1 switch #: 11
pipe 2 switch #: 14
pipe 3 switch #: 4

input #: 132
switch input port #: 4
pipe 0 switch #: 8
pipe 1 switch #: 12
pipe 2 switch #: 0
pipe 3 switch #: 11

input #: 133
switch input port #: 5
pipe 0 switch #: 8
pipe 1 switch #: 13
pipe 2 switch #: 2
pipe 3 switch #: 15

input #: 6
switch input port #: 6
pipe 0 switch #: 0
pipe 1 switch #: 6
pipe 2 switch #: 12
pipe 3 switch #: 11

input #: 7
switch input port #: 7
pipe 0 switch #: 0
pipe 1 switch #: 7
pipe 2 switch #: 14
pipe 3 switch #: 15

input #: 8
switch input port #: 8
pipe 0 switch #: 0
pipe 1 switch #: 8
pipe 2 switch #: 3
pipe 3 switch #: 6

input #: 9
switch input port #: 9
pipe 0 switch #: 0
pipe 1 switch #: 9
pipe 2 switch #: 1
pipe 3 switch #: 2

input #: 10
switch input port #: 10
pipe 0 switch #: 0
pipe 1 switch #: 10
pipe 2 switch #: 7
pipe 3 switch #: 14

input #: 11
switch input port #: 11
pipe 0 switch #: 0
pipe 1 switch #: 11
pipe 2 switch #: 5
pipe 3 switch #: 10

input #: 134
switch input port #: 6
pipe 0 switch #: 8
pipe 1 switch #: 14
pipe 2 switch #: 4
pipe 3 switch #: 3

input #: 135
switch input port #: 7
pipe 0 switch #: 8
pipe 1 switch #: 15
pipe 2 switch #: 6
pipe 3 switch #: 7

input #: 136
switch input port #: 8
pipe 0 switch #: 8
pipe 1 switch #: 0
pipe 2 switch #: 11
pipe 3 switch #: 14

input #: 137
switch input port #: 9
pipe 0 switch #: 8
pipe 1 switch #: 1
pipe 2 switch #: 9
pipe 3 switch #: 10

input #: 138
switch input port #: 10
pipe 0 switch #: 8
pipe 1 switch #: 2
pipe 2 switch #: 15
pipe 3 switch #: 6

input #: 139
switch input port #: 11
pipe 0 switch #: 8
pipe 1 switch #: 3
pipe 2 switch #: 13
pipe 3 switch #: 2

input #: 12
switch input port #:12
pipe 0 switch #: 0
pipe 1 switch #: 12
pipe 2 switch #: 11
pipe 3 switch #: 5

input #: 13
switch input port #: 13
pipe 0 switch #: 0
pipe 1 switch #: 13
pipe 2 switch #: 9
pipe 3 switch #: 1

input #: 14
switch input port #: 14
pipe 0 switch #: 0
pipe 1 switch #: 14
pipe 2 switch #: 15
pipe 3 switch #: 13

input #: 15
switch input port #: 15
pipe 0 switch #: 0
pipe 1 switch #: 15
pipe 2 switch #: 13
pipe 3 switch #: 9

input #: 16
switch input port #: 0
pipe 0 switch #: 1
pipe 1 switch #: 1
pipe 2 switch #: 1
pipe 3 switch #: 1

input #: 17
switch input port #: 1
pipe 0 switch #: 1
pipe 1 switch #: 0
pipe 2 switch #: 3
pipe 3 switch #: 5

input #: 140
switch input port #: 12
pipe 0 switch #: 8
pipe 1 switch #: 4
pipe 2 switch #: 3
pipe 3 switch #: 13

input #: 141
switch input port #: 13
pipe 0 switch #: 8
pipe 1 switch #: 5
pipe 2 switch #: 1
pipe 3 switch #: 9

input #: 142
switch input port #: 14
pipe 0 switch #: 8
pipe 1 switch #: 6
pipe 2 switch #: 7
pipe 3 switch #: 5

input #: 143
switch input port #: 15
pipe 0 switch #: 8
pipe 1 switch #: 7
pipe 2 switch #: 5
pipe 3 switch #: 1

input #: 144
switch input port #: 0
pipe 0 switch #: 9
pipe 1 switch #: 9
pipe 2 switch #: 9
pipe 3 switch #: 9

input #: 145
switch input port #:1
pipe 0 switch #: 9
pipe 1 switch #: 8
pipe 2 switch #: 11
pipe 3 switch #: 13

input #: 18
switch input port #: 2
pipe 0 switch #: 1
pipe 1 switch #: 3
pipe 2 switch #: 5
pipe 3 switch #: 9

input #: 19
switch input port #: 3
pipe 0 switch #: 1
pipe 1 switch #: 2
pipe 2 switch #: 7
pipe 3 switch #: 13

input #: 20
switch input port #: 4
pipe 0 switch #: 1
pipe 1 switch #: 5
pipe 2 switch #: 9
pipe 3 switch #: 2

input #: 21
switch input port #: 5
pipe 0 switch #: 1
pipe 1 switch #: 4
pipe 2 switch #: 11
pipe 3 switch #: 6

input #: 22
switch input port #: 6
pipe 0 switch #: 1
pipe 1 switch #: 7
pipe 2 switch #: 13
pipe 3 switch #: 10

input #: 23
switch input port #: 7
pipe 0 switch #: 1
pipe 1 switch #: 6
pipe 2 switch #: 15
pipe 3 switch #: 14

input #: 146
switch input port #: 2
pipe 0 switch #: 9
pipe 1 switch #: 11
pipe 2 switch #: 13
pipe 3 switch #: 1

input #: 147
switch input port #: 3
pipe 0 switch #: 9
pipe 1 switch #: 10
pipe 2 switch #: 15
pipe 3 switch #: 5

input #: 148
switch input port #: 4
pipe 0 switch #: 9
pipe 1 switch #: 13
pipe 2 switch #: 1
pipe 3 switch #: 10

input #: 149
switch input port #: 5
pipe 0 switch #: 9
pipe 1 switch #: 12
pipe 2 switch #: 3
pipe 3 switch #: 14

input #: 150
switch input port #: 6
pipe 0 switch #: 9
pipe 1 switch #: 15
pipe 2 switch #: 5
pipe 3 switch #: 2

input #: 151
switch input port #: 7
pipe 0 switch #: 9
pipe 1 switch #: 14
pipe 2 switch #: 7
pipe 3 switch #: 6

input #: 24
switch input port #: 8
pipe 0 switch #: 1
pipe 1 switch #: 9
pipe 2 switch #: 2
pipe 3 switch #: 7

input #: 25
switch input port #: 9
pipe 0 switch #: 1
pipe 1 switch #: 8
pipe 2 switch #: 0
pipe 3 switch #: 3

input #: 26
switch input port #: 10
pipe 0 switch #: 1
pipe 1 switch #: 11
pipe 2 switch #: 6
pipe 3 switch #: 15

input #: 27
switch input port #: 11
pipe 0 switch #: 1
pipe 1 switch #: 10
pipe 2 switch #: 4
pipe 3 switch #: 11

input #: 28
switch input port #: 12
pipe 0 switch #: 1
pipe 1 switch #: 13
pipe 2 switch #: 10
pipe 3 switch #: 4

input #: 29
switch input port #: 13
pipe 0 switch #: 1
pipe 1 switch #: 12
pipe 2 switch #: 8
pipe 3 switch #: 0

input #: 152
switch input port #: 8
pipe 0 switch #: 9
pipe 1 switch #: 1
pipe 2 switch #: 10
pipe 3 switch #: 15

input #: 153
switch input port #: 9
pipe 0 switch #: 9
pipe 1 switch #: 0
pipe 2 switch #: 8
pipe 3 switch #: 11

input #: 154
switch input port #: 10
pipe 0 switch #: 9
pipe 1 switch #: 3
pipe 2 switch #: 14
pipe 3 switch #: 7

input #: 155
switch input port #: 11
pipe 0 switch #: 9
pipe 1 switch #: 2
pipe 2 switch #: 12
pipe 3 switch #: 3

input #:156
switch input port #: 12
pipe 0 switch #: 9
pipe 1 switch #: 5
pipe 2 switch #: 2
pipe 3 switch #: 12

input #: 157
switch input port #: 13
pipe 0 switch #: 9
pipe 1 switch #: 4
pipe 2 switch #: 0
pipe 3 switch #: 8

input #: 30
switch input port #: 14
pipe 0 switch #: 1
pipe 1 switch #: 15
pipe 2 switch #: 14
pipe 3 switch #: 12

input #: 31
switch input port #: 15
pipe 0 switch #: 1
pipe 1 switch #: 14
pipe 2 switch #: 12
pipe 3 switch #: 8
input #: 32
switch input port #: 0
pipe 0 switch #: 2
pipe 1 switch #: 2
pipe 2 switch #: 2
pipe 3 switch #: 2

input #: 33
switch input port #: 1
pipe 0 switch #: 2
pipe 1 switch #: 3
pipe 2 switch #: 0
pipe 3 switch #: 6

input #: 34
switch input port #: 2
pipe 0 switch #: 2
pipe 1 switch #: 0
pipe 2 switch #: 6
pipe 3 switch #: 10

input #: 35
switch input port #: 3
pipe 0 switch #: 2
pipe 1 switch #: 1
pipe 2 switch #: 4
pipe 3 switch #: 14

input #: 158
switch input port #: 14
pipe 0 switch #: 9
pipe 1 switch #: 7
pipe 2 switch #: 6
pipe 3 switch #: 4

input #: 159
switch input port #: 15
pipe 0 switch #: 9
pipe 1 switch #: 6
pipe 2 switch #: 4
pipe 3 switch #: 0
input #: 160
switch input port #: 0
pipe 0 switch #: 10
pipe 1 switch #: 10
pipe 2 switch #: 10
pipe 3 switch #: 10

input #: 161
switch input port #: 1
pipe 0 switch #: 10
pipe 1 switch #: 11
pipe 2 switch #: 8
pipe 3 switch #: 14

input #: 162
switch input port #: 2
pipe 0 switch #: 10
pipe 1 switch #: 8
pipe 2 switch #: 14
pipe 3 switch #: 2

input #: 163
switch input port #: 3
pipe 0 switch #: 10
pipe 1 switch #: 9
pipe 2 switch #: 12
pipe 3 switch #: 6

input #: 36
switch input port #: 4
pipe 0 switch #: 2
pipe 1 switch #: 6
pipe 2 switch #: 10
pipe 3 switch #: 1

input #: 37
switch input port #: 5
pipe 0 switch #: 2
pipe 1 switch #: 7
pipe 2 switch #: 8
pipe 3 switch #: 5

input #: 38
switch input port #: 6
pipe 0 switch #: 2
pipe 1 switch #: 4
pipe 2 switch #: 14
pipe 3 switch #: 9

input #: 39
switch input port #: 7
pipe 0 switch #: 2
pipe 1 switch #: 5
pipe 2 switch #: 12
pipe 3 switch #: 13

input #: 40
switch input port #: 8
pipe 0 switch #: 2
pipe 1 switch #: 10
pipe 2 switch #: 1
pipe 3 switch #: 4

input #: 41
switch input port #: 9
pipe 0 switch #: 2
pipe 1 switch #: 11
pipe 2 switch #: 3
pipe 3 switch #: 0

input #: 164
switch input port #: 4
pipe 0 switch #: 10
pipe 1 switch #: 14
pipe 2 switch #: 2
pipe 3 switch #: 9

input #: 165
switch input port #: 5
pipe 0 switch #: 10
pipe 1 switch #: 15
pipe 2 switch #: 0
pipe 3 switch #: 13

input #: 166
switch input port #: 6
pipe 0 switch #: 10
pipe 1 switch #: 12
pipe 2 switch #: 6
pipe 3 switch #: 1

input #: 167
switch input port #: 7
pipe 0 switch #: 10
pipe 1 switch #: 13
pipe 2 switch #: 4
pipe 3 switch #: 5

input #: 168
switch input port #: 8
pipe 0 switch #: 10
pipe 1 switch #: 2
pipe 2 switch #: 9
pipe 3 switch #: 12

input #: 169
switch input port #: 9
pipe 0 switch #: 10
pipe 1 switch #: 3
pipe 2 switch #: 11
pipe 3 switch #: 8

input #: 42
switch input port #: 10
pipe 0 switch #: 2
pipe 1 switch #: 8
pipe 2 switch #: 5
pipe 3 switch #: 12

input #: 43
switch input port #: 11
pipe 0 switch #: 2
pipe 1 switch #: 9
pipe 2 switch #: 7
pipe 3 switch #: 8

input #: 44
switch input port #: 12
pipe 0 switch #: 2
pipe 1 switch #: 14
pipe 2 switch #: 9
pipe 3 switch #: 7

input #: 45
switch input port #: 13
pipe 0 switch #: 2
pipe 1 switch #: 15
pipe 2 switch #: 11
pipe 3 switch #: 3

input #: 46
switch input port #: 14
pipe 0 switch #: 2
pipe 1 switch #: 12
pipe 2 switch #: 13
pipe 3 switch #: 15

input #: 47
switch input port #: 15
pipe 0 switch #: 2
pipe 1 switch #: 13
pipe 2 switch #: 15
pipe 3 switch #: 11

input #: 170
switch input port #: 10
pipe 0 switch #: 10
pipe 1 switch #: 0
pipe 2 switch #: 13
pipe 3 switch #: 4

input #: 171
switch input port #: 11
pipe 0 switch #: 10
pipe 1 switch #: 1
pipe 2 switch #: 15
pipe 3 switch #: 0

input #: 172
switch input port #: 12
pipe 0 switch #: 10
pipe 1 switch #: 6
pipe 2 switch #: 1
pipe 3 switch #: 15

input #: 173
switch input port #: 13
pipe 0 switch #: 10
pipe 1 switch #: 7
pipe 2 switch #: 3
pipe 3 switch #: 11

input #: 174
switch input port #: 14
pipe 0 switch #: 10
pipe 1 switch #: 4
pipe 2 switch #: 5
pipe 3 switch #: 7

input #: 175
switch input port #: 15
pipe 0 switch #: 10
pipe 1 switch #: 5
pipe 2 switch #: 7
pipe 3 switch #: 3

input #: 48
switch input port #: 0
pipe 0 switch #: 3
pipe 1 switch #: 3
pipe 2 switch #: 3
pipe 3 switch #: 3

input #: 49
switch input port #: 1
pipe 0 switch #: 3
pipe 1 switch #: 2
pipe 2 switch #: 1
pipe 3 switch #: 7

input #: 50
switch input port #: 2
pipe 0 switch #: 3
pipe 1 switch #: 1
pipe 2 switch #: 7
pipe 3 switch #: 11

input #: 51
switch input port #: 3
pipe 0 switch #: 3
pipe 1 switch #: 0
pipe 2 switch #: 5
pipe 3 switch #: 15

input #: 52
switch input port #: 4
pipe 0 switch #: 3
pipe 1 switch #: 7
pipe 2 switch #: 11
pipe 3 switch #: 0

input #: 53
switch input port #: 5
pipe 0 switch #: 3
pipe 1 switch #: 6
pipe 2 switch #: 9
pipe 3 switch #: 4

input #: 176
switch input port #: 0
pipe 0 switch #: 11
pipe 1 switch #: 11
pipe 2 switch #: 11
pipe 3 switch #: 11

input #: 177
switch input port #: 1
pipe 0 switch #: 11
pipe 1 switch #: 10
pipe 2 switch #: 9
pipe 3 switch #: 15

input #: 178
switch input port #: 2
pipe 0 switch #: 11
pipe 1 switch #: 9
pipe 2 switch #: 15
pipe 3 switch #: 3

input #: 179
switch input port #: 3
pipe 0 switch #: 11
pipe 1 switch #: 8
pipe 2 switch #: 13
pipe 3 switch #: 7

input #: 180
switch input port #: 4
pipe 0 switch #: 11
pipe 1 switch #: 15
pipe 2 switch #: 3
pipe 3 switch #: 8

input #: 181
switch input port #: 5
pipe 0 switch #: 11
pipe 1 switch #: 14
pipe 2 switch #: 1
pipe 3 switch #: 12

input #: 54
switch input port #: 6
pipe 0 switch #: 3
pipe 1 switch #: 5
pipe 2 switch #: 15
pipe 3 switch #: 8

input #: 55
switch input port #: 7
pipe 0 switch #: 3
pipe 1 switch #: 4
pipe 2 switch #: 13
pipe 3 switch #: 12

input #: 56
switch input port #: 8
pipe 0 switch #: 3
pipe 1 switch #: 11
pipe 2 switch #: 0
pipe 3 switch #: 5

input #: 57
switch input port #: 9
pipe 0 switch #: 3
pipe 1 switch #: 10
pipe 2 switch #: 2
pipe 3 switch #: 1

input #: 58
switch input port #: 10
pipe 0 switch #: 3
pipe 1 switch #: 9
pipe 2 switch #: 4
pipe 3 switch #: 13

input #: 59
switch input port #: 11
pipe 0 switch #: 3
pipe 1 switch #: 8
pipe 2 switch #: 6
pipe 3 switch #: 9

input #: 182
switch input port #: 6
pipe 0 switch #: 11
pipe 1 switch #: 13
pipe 2 switch #: 7
pipe 3 switch #: 0

input #: 183
switch input port #: 7
pipe 0 switch #: 11
pipe 1 switch #: 12
pipe 2 switch #: 5
pipe 3 switch #: 4

input #: 184
switch input port #: 8
pipe 0 switch #: 11
pipe 1 switch #: 3
pipe 2 switch #: 8
pipe 3 switch #: 13

input #: 185
switch input port #: 9
pipe 0 switch #: 11
pipe 1 switch #: 2
pipe 2 switch #: 10
pipe 3 switch #: 9

input #: 186
switch input port #: 10
pipe 0 switch #: 11
pipe 1 switch #: 1
pipe 2 switch #: 12
pipe 3 switch #: 5

input #: 187
switch input port #: 11
pipe 0 switch #: 11
pipe 1 switch #: 0
pipe 2 switch #: 14
pipe 3 switch #: 1

input #: 60
switch input port #: 12
pipe 0 switch #: 3
pipe 1 switch #: 15
pipe 2 switch #: 8
pipe 3 switch #: 6

input #: 61
switch input port #: 13
pipe 0 switch #: 3
pipe 1 switch #: 14
pipe 2 switch #: 10
pipe 3 switch #: 2

input #: 62
switch input port #: 14
pipe 0 switch #: 3
pipe 1 switch #: 13
pipe 2 switch #: 12
pipe 3 switch #: 14

input #: 63
switch input port #: 15
pipe 0 switch #: 3
pipe 1 switch #: 12
pipe 2 switch #: 14
pipe 3 switch #: 10

input #: 64
switch input port #: 0
pipe 0 switch #: 4
pipe 1 switch #: 4
pipe 2 switch #: 4
pipe 3 switch #: 4

input #: 65
switch input port #: 1
pipe 0 switch #: 4
pipe 1 switch #: 5
pipe 2 switch #: 6
pipe 3 switch #: 0

input #: 188
switch input port #: 12
pipe 0 switch #: 11
pipe 1 switch #: 7
pipe 2 switch #: 0
pipe 3 switch #: 14

input #: 189
switch input port #: 13
pipe 0 switch #: 11
pipe 1 switch #: 6
pipe 2 switch #: 2
pipe 3 switch #: 10

input #: 190
switch input port #: 14
pipe 0 switch #: 11
pipe 1 switch #: 5
pipe 2 switch #: 4
pipe 3 switch #: 6

input #: 191
switch input port #: 15
pipe 0 switch #: 11
pipe 1 switch #: 4
pipe 2 switch #: 6
pipe 3 switch #: 2

input #: 192
switch input port #: 0
pipe 0 switch #: 12
pipe 1 switch #: 12
pipe 2 switch #: 12
pipe 3 switch #: 12

input #: 193
switch input port #: 1
pipe 0 switch #: 12
pipe 1 switch #: 13
pipe 2 switch #: 14
pipe 3 switch #: 8

input #: 66
switch input port #: 2
pipe 0 switch #: 4
pipe 1 switch #: 6
pipe 2 switch #: 0
pipe 3 switch #: 12

input #: 67
switch input port #: 3
pipe 0 switch #: 4
pipe 1 switch #: 7
pipe 2 switch #: 2
pipe 3 switch #: 8

input #: 68
switch input port #: 4
pipe 0 switch #: 4
pipe 1 switch #: 0
pipe 2 switch #: 12
pipe 3 switch #: 7

input #: 69
switch input port #: 5
pipe 0 switch #: 4
pipe 1 switch #: 1
pipe 2 switch #: 14
pipe 3 switch #: 3

input #: 70
switch input port #: 6
pipe 0 switch #: 4
pipe 1 switch #: 2
pipe 2 switch #: 8
pipe 3 switch #: 15

input #: 71
switch inp

input #: 194
switch input port #: 2
pipe 0 switch #: 12
pipe 1 switch #: 14
pipe 2 switch #: 8
pipe 3 switch #: 4

input #: 195
switch input port #: 3
pipe 0 switch #: 12
pipe 1 switch #: 15
pipe 2 switch #: 10
pipe 3 switch #: 0

input #: 196
switch input port #: 4
pipe 0 switch #: 12
pipe 1 switch #: 8
pipe 2 switch #: 4
pipe 3 switch #: 15

input #: 197
switch input port #: 5
pipe 0 switch #: 12
pipe 1 switch #: 9
pipe 2 switch #: 6
pipe 3 switch #: 11

input #: 198
switch input port #: 6
pipe 0 switch #: 12
pipe 1 switch #: 10
pipe 2 switch #: 0
pipe 3 switch #: 7

input #: 199

input #: 72
switch input port #: 8
pipe 0 switch #: 4
pipe 1 switch #: 12
pipe 2 switch #: 7
pipe 3 switch #: 2

input #: 73
switch input port #: 9
pipe 0 switch #: 4
pipe 1 switch #: 13
pipe 2 switch #: 5
pipe 3 switch #: 6

input #: 74
switch input port #: 10
pipe 0 switch #: 4
pipe 1 switch #: 14
pipe 2 switch #: 3
pipe 3 switch #: 10

input #: 75
switch input port #: 11
pipe 0 switch #: 4
pipe 1 switch #: 15
pipe 2 switch #: 1
pipe 3 switch #: 14

input #: 76
switch input port #: 12
pipe 0 switch #: 4
pipe 1 switch #: 8
pipe 2 switch #: 15
pipe 3 switch #: 1

input #: 77
switch input port #: 13
pipe 0 switch #: 4
pipe 1 switch #: 9
pipe 2 switch #: 13
pipe 3 switch #: 5

input #: 200
switch input port #: 8
pipe 0 switch #: 12
pipe 1 switch #: 4
pipe 2 switch #: 15
pipe 3 switch #: 10

input #: 201
switch input port #: 9
pipe 0 switch #: 12
pipe 1 switch #: 5
pipe 2 switch #: 13
pipe 3 switch #: 14

input #: 202
switch input port #: 10
pipe 0 switch #: 12
pipe 1 switch #: 6
pipe 2 switch #: 11
pipe 3 switch #: 2

input #: 203
switch input port #: 11
pipe 0 switch #: 12
pipe 1 switch #: 7
pipe 2 switch #: 9
pipe 3 switch #: 6

input #: 204
switch input port #: 12
pipe 0 switch #: 12
pipe 1 switch #: 0
pipe 2 switch #: 7
pipe 3 switch #: 9

input #: 205
switch input port #: 13
pipe 0 switch #: 12
pipe 1 switch #: 1
pipe 2 switch #: 5
pipe 3 switch #: 13

input #: 78
switch input port #: 14
pipe 0 switch #: 4
pipe 1 switch #: 10
pipe 2 switch #: 11
pipe 3 switch #: 9

input #: 79
switch input port #: 15
pipe 0 switch #: 4
pipe 1 switch #: 11
pipe 2 switch #: 9
pipe 3 switch #: 13

input #: 80
switch input port #: 0
pipe 0 switch #: 5
pipe 1 switch #: 5
pipe 2 switch #: 5
pipe 3 switch #: 5

input #: 81
switch input port #: 1
pipe 0 switch #: 5
pipe 1 switch #: 4
pipe 2 switch #: 7
pipe 3 switch #: 1

input #: 82
switch input port #: 2
pipe 0 switch #: 5
pipe 1 switch #: 7
pipe 2 switch #: 1
pipe 3 switch #: 13

input #: 83
switch input port #: 3
pipe 0 switch #: 5
pipe 1 switch #: 6
pipe 2 switch #: 3
pipe 3 switch #: 9

input #: 206
switch input port #: 14
pipe 0 switch #: 12
pipe 1 switch #: 2
pipe 2 switch #: 3
pipe 3 switch #: 1

input #: 207
switch input port #: 15
pipe 0 switch #: 12
pipe 1 switch #: 3
pipe 2 switch #: 1
pipe 3 switch #: 5

input #: 208
switch input port #: 0
pipe 0 switch #: 13
pipe 1 switch #: 13
pipe 2 switch #: 13
pipe 3 switch #: 13

input #: 209
switch input port #: 1
pipe 0 switch #: 13
pipe 1 switch #: 12
pipe 2 switch #: 15
pipe 3 switch #: 9

input #: 210
switch input port #: 2
pipe 0 switch #: 13
pipe 1 switch #: 15
pipe 2 switch #: 9
pipe 3 switch #: 5

input #: 211
switch input port #: 3
pipe 0 switch #: 13
pipe 1 switch #: 14
pipe 2 switch #: 11
pipe 3 switch #: 1

input #: 84
switch input port #: 4
pipe 0 switch #: 5
pipe 1 switch #: 1
pipe 2 switch #: 13
pipe 3 switch #: 6

input #: 85
switch input port #: 5
pipe 0 switch #: 5
pipe 1 switch #: 0
pipe 2 switch #: 15
pipe 3 switch #: 2

input #: 86
switch input port #: 6
pipe 0 switch #: 5
pipe 1 switch #: 3
pipe 2 switch #: 9
pipe 3 switch #: 14

input #: 87
switch input port #: 7
pipe 0 switch #: 5
pipe 1 switch #: 2
pipe 2 switch #: 11
pipe 3 switch #: 10

input #: 88
switch input port #: 8
pipe 0 switch #: 5
pipe 1 switch #: 13
pipe 2 switch #: 6
pipe 3 switch #: 3

input #: 89
switch input port #: 9
pipe 0 switch #: 5
pipe 1 switch #: 12
pipe 2 switch #: 4
pipe 3 switch #: 7

input #: 212
switch input port #: 4
pipe 0 switch #: 13
pipe 1 switch #: 9
pipe 2 switch #: 5
pipe 3 switch #: 14

input #: 213
switch input port #: 5
pipe 0 switch #: 13
pipe 1 switch #: 8
pipe 2 switch #: 7
pipe 3 switch #: 10

input #: 214
switch input port #: 6
pipe 0 switch #: 13
pipe 1 switch #: 11
pipe 2 switch #: 1
pipe 3 switch #: 6

input #: 215
switch input port #: 7
pipe 0 switch #: 13
pipe 1 switch #: 10
pipe 2 switch #: 3
pipe 3 switch #: 2

input #: 216
switch input port #: 8
pipe 0 switch #: 13
pipe 1 switch #: 5
pipe 2 switch #: 14
pipe 3 switch #: 11

input #: 217
switch input port #: 9
pipe 0 switch #: 13
pipe 1 switch #: 4
pipe 2 switch #: 12
pipe 3 switch #: 15

input #: 90
switch input port #: 10
pipe 0 switch #: 5
pipe 1 switch #: 15
pipe 2 switch #: 2
pipe 3 switch #: 11

input #: 91
switch input port #: 11
pipe 0 switch #: 5
pipe 1 switch #: 14
pipe 2 switch #: 0
pipe 3 switch #: 15

input #: 92
switch input port #: 12
pipe 0 switch #: 5
pipe 1 switch #: 9
pipe 2 switch #: 14
pipe 3 switch #: 0
input #: 93
switch input port #: 13
pipe 0 switch #: 5
pipe 1 switch #: 8
pipe 2 switch #: 12
pipe 3 switch #: 4

input #: 94
switch input port #: 14
pipe 0 switch #: 5
pipe 1 switch #: 11
pipe 2 switch #: 10
pipe 3 switch #: 8

input #: 95
switch input port #: 15
pipe 0 switch #: 5
pipe 1 switch #: 10
pipe 2 switch #: 8
pipe 3 switch #: 12

input #: 218
switch input port #: 10
pipe 0 switch #: 13
pipe 1 switch #: 7
pipe 2 switch #: 10
pipe 3 switch #: 3

input #: 219
switch input port #: 11
pipe 0 switch #: 13
pipe 1 switch #: 6
pipe 2 switch #: 8
pipe 3 switch #: 7

input #: 220
switch input port #: 12
pipe 0 switch #: 13
pipe 1 switch #: 1
pipe 2 switch #: 6
pipe 3 switch #: 8
input #: 221
switch input port #: 13
pipe 0 switch #: 13
pipe 1 switch #: 0
pipe 2 switch #: 4
pipe 3 switch #: 12

input #: 222
switch input port #: 14
pipe 0 switch #: 13
pipe 1 switch #: 3
pipe 2 switch #: 2
pipe 3 switch #: 0

input #: 223
switch input port #: 15
pipe 0 switch #: 13
pipe 1 switch #: 2
pipe 2 switch #: 0
pipe 3 switch #: 4

input #: 96
switch input port #: 0
pipe 0 switch #: 6
pipe 1 switch #: 6
pipe 2 switch #: 6
pipe 3 switch #: 6

input #: 97
switch input port #: 1
pipe 0 switch #: 6
pipe 1 switch #: 7
pipe 2 switch #: 4
pipe 3 switch #: 2

input #: 98
switch input port #: 2
pipe 0 switch #: 6
pipe 1 switch #: 4
pipe 2 switch #: 2
pipe 3 switch #: 14

input #: 99
switch input port #: 3
pipe 0 switch #: 6
pipe 1 switch #: 5
pipe 2 switch #: 0
pipe 3 switch #: 10

input #: 100
switch input port #: 4
pipe 0 switch #: 6
pipe 1 switch #: 2
pipe 2 switch #: 14
pipe 3 switch #: 5

input #: 101
switch input port #: 5
pipe 0 switch #: 6
pipe 1 switch #: 3
pipe 2 switch #: 12
pipe 3 switch #: 1

input #: 224
switch input port #: 0
pipe 0 switch #: 14
pipe 1 switch #: 14
pipe 2 switch #: 14
pipe 3 switch #: 14

input #: 225
switch input port #: 1
pipe 0 switch #: 14
pipe 1 switch #: 15
pipe 2 switch #: 12
pipe 3 switch #: 10

input #: 226
switch input port #: 2
pipe 0 switch #: 14
pipe 1 switch #: 12
pipe 2 switch #: 10
pipe 3 switch #: 6

input #: 227
switch input port #: 3
pipe 0 switch #: 14
pipe 1 switch #: 13
pipe 2 switch #: 8
pipe 3 switch #: 2

input #: 228
switch input port #: 4
pipe 0 switch #: 14
pipe 1 switch #: 10
pipe 2 switch #: 6
pipe 3 switch #: 13

input #: 229
switch input port #: 5
pipe 0 switch #: 14
pipe 1 switch #: 11
pipe 2 switch #: 4
pipe 3 switch #: 9

input #: 102
switch input port #: 6
pipe 0 switch #: 6
pipe 1 switch #: 0
pipe 2 switch #: 10
pipe 3 switch #: 13

input #: 103
switch input port #: 7
pipe 0 switch #: 6
pipe 1 switch #: 1
pipe 2 switch #: 8
pipe 3 switch #: 9

input #: 104
switch input port #: 8
pipe 0 switch #: 6
pipe 1 switch #: 14
pipe 2 switch #: 5
pipe 3 switch #: 0

input #: 105
switch input port #: 9
pipe 0 switch #: 6
pipe 1 switch #: 15
pipe 2 switch #: 7
pipe 3 switch #: 4

input #: 106
switch input port #: 10
pipe 0 switch #: 6
pipe 1 switch #: 12
pipe 2 switch #: 1
pipe 3 switch #: 8
input #: 107
switch input port #: 11
pipe 0 switch #: 6
pipe 1 switch #: 13
pipe 2 switch #: 3
pipe 3 switch #: 12

input #: 230
switch input port #: 6
pipe 0 switch #: 14
pipe 1 switch #: 8
pipe 2 switch #: 2
pipe 3 switch #: 5

input #: 231
switch input port #: 7
pipe 0 switch #: 14
pipe 1 switch #: 9
pipe 2 switch #: 0
pipe 3 switch #: 1

input #: 232
switch input port #: 8
pipe 0 switch #: 14
pipe 1 switch #: 6
pipe 2 switch #: 13
pipe 3 switch #: 8

input #: 233
switch input port #: 9
pipe 0 switch #: 14
pipe 1 switch #: 7
pipe 2 switch #: 15
pipe 3 switch #: 12

input #: 234
switch input port #: 10
pipe 0 switch #: 14
pipe 1 switch #: 4
pipe 2 switch #: 9
pipe 3 switch #: 0
input #: 235
switch input port #: 11
pipe 0 switch #: 14
pipe 1 switch #: 5
pipe 2 switch #: 11
pipe 3 switch #: 4

input #: 108
switch input port #: 12
pipe 0 switch #: 6
pipe 1 switch #: 10
pipe 2 switch #: 13
pipe 3 switch #: 3

input #: 109
switch input port #: 13
pipe 0 switch #: 6
pipe 1 switch #: 11
pipe 2 switch #: 15
pipe 3 switch #: 7

input #: 110
switch input port #: 14
pipe 0 switch #: 6
pipe 1 switch #: 8
pipe 2 switch #: 9
pipe 3 switch #: 11

input #: 111
switch input port #: 15
pipe 0 switch #: 6
pipe 1 switch #: 9
pipe 2 switch #: 11
pipe 3 switch #: 15

input #: 112
switch input port #: 0
pipe 0 switch #: 7
pipe 1 switch #: 7
pipe 2 switch #: 7
pipe 3 switch #: 7

input #: 113
switch input port #: 1
pipe 0 switch #: 7
pipe 1 switch #: 6
pipe 2 switch #: 5
pipe 3 switch #: 3

input #: 236
switch input port #: 12
pipe 0 switch #: 14
pipe 1 switch #: 2
pipe 2 switch #: 5
pipe 3 switch #: 11

input #: 237
switch input port #: 13
pipe 0 switch #: 14
pipe 1 switch #: 3
pipe 2 switch #: 7
pipe 3 switch #: 15

input #: 238
switch input port #: 14
pipe 0 switch #: 14
pipe 1 switch #: 0
pipe 2 switch #: 1
pipe 3 switch #: 3

input #: 239
switch input port #: 15
pipe 0 switch #: 14
pipe 1 switch #: 1
pipe 2 switch #: 3
pipe 3 switch #: 7

input #: 240
switch input port #: 0
pipe 0 switch #: 15
pipe 1 switch #: 15
pipe 2 switch #: 15
pipe 3 switch #: 15

input #: 241
switch input port #: 1
pipe 0 switch #: 15
pipe 1 switch #: 14
pipe 2 switch #: 13
pipe 3 switch #: 11

input #: 114
switch input port #: 2
pipe 0 switch #: 7
pipe 1 switch #: 5
pipe 2 switch #: 3
pipe 3 switch #: 15

input #: 115
switch input port #: 3
pipe 0 switch #: 7
pipe 1 switch #: 4
pipe 2 switch #: 1
pipe 3 switch #: 11

input #: 116
switch input port #: 4
pipe 0 switch #: 7
pipe 1 switch #: 3
pipe 2 switch #: 15
pipe 3 switch #: 4

input #: 117
switch input port #: 5
pipe 0 switch #: 7
pipe 1 switch #: 2
pipe 2 switch #: 13
pipe 3 switch #: 0

input #: 118
switch input port #: 6
pipe 0 switch #: 7
pipe 1 switch #: 1
pipe 2 switch #: 11
pipe 3 switch #: 12

input #: 119
switch input port #: 7
pipe 0 switch #: 7
pipe 1 switch #: 0
pipe 2 switch #: 9
pipe 3 switch #: 8

input #: 242
switch input port #: 2
pipe 0 switch #: 15
pipe 1 switch #: 13
pipe 2 switch #: 11
pipe 3 switch #: 7

input #: 243
switch input port #: 3
pipe 0 switch #: 15
pipe 1 switch #: 12
pipe 2 switch #: 9
pipe 3 switch #: 3

input #: 244
switch input port #: 4
pipe 0 switch #: 15
pipe 1 switch #: 11
pipe 2 switch #: 7
pipe 3 switch #: 12

input #: 245
switch input port #: 5
pipe 0 switch #: 15
pipe 1 switch #: 10
pipe 2 switch #: 5
pipe 3 switch #: 8

input #: 246
switch input port #: 6
pipe 0 switch #: 15
pipe 1 switch #: 9
pipe 2 switch #: 3
pipe 3 switch #: 4

input #: 247
switch input port #: 7
pipe 0 switch #: 15
pipe 1 switch #: 8
pipe 2 switch #: 1
pipe 3 switch #: 0

input #: 120
switch input port #: 8
pipe 0 switch #: 7
pipe 1 switch #: 15
pipe 2 switch #: 4
pipe 3 switch #: 1
input #: 121
switch input port #: 9
pipe 0 switch #: 7
pipe 1 switch #: 14
pipe 2 switch #: 6
pipe 3 switch #: 5

input #: 122
switch input port #: 10
pipe 0 switch #: 7
pipe 1 switch #: 13
pipe 2 switch #: 0
pipe 3 switch #: 9

input #: 123
switch input port #: 11
pipe 0 switch #: 7
pipe 1 switch #: 12
pipe 2 switch #: 2
pipe 3 switch #: 13

input #: 124
switch input port #: 12
pipe 0 switch #: 7
pipe 1 switch #: 11
pipe 2 switch #: 12
pipe 3 switch #: 2

input #: 125
switch input port #: 13
pipe 0 switch #: 7
pipe 1 switch #: 10
pipe 2 switch #: 14
pipe 3 switch #: 6

input #: 248
switch input port #: 8
pipe 0 switch #: 15
pipe 1 switch #: 7
pipe 2 switch #: 12
pipe 3 switch #: 9
input #: 249
switch input port #: 9
pipe 0 switch #: 15
pipe 1 switch #: 6
pipe 2 switch #: 14
pipe 3 switch #: 13

input #: 250
switch input port #: 10
pipe 0 switch #: 15
pipe 1 switch #: 5
pipe 2 switch #: 8
pipe 3 switch #: 1

input #: 251
switch input port #: 11
pipe 0 switch #: 15
pipe 1 switch #: 4
pipe 2 switch #: 10
pipe 3 switch #: 5

input #: 252
switch input port #: 12
pipe 0 switch #: 15
pipe 1 switch #: 3
pipe 2 switch #: 4
pipe 3 switch #: 10

input #: 253
switch input port #: 13
pipe 0 switch #: 15
pipe 1 switch #: 2
pipe 2 switch #: 6
pipe 3 switch #: 14

input #: 126
switch input port #: 14
pipe 0 switch #: 7
pipe 1 switch #: 9
pipe 2 switch #: 8
pipe 3 switch #: 10

input #: 254
switch input port #: 14
pipe 0 switch #: 15
pipe 1 switch #: 1
pipe 2 switch #: 0
pipe 3 switch #: 2

input #: 127
switch input port #: 15
pipe 0 switch #: 7
pipe 1 switch #: 8
pipe 2 switch #: 10
pipe 3 switch #: 14

input #: 255
switch input port #: 15
pipe 0 switch #: 15
pipe 1 switch #: 0
pipe 2 switch #: 2
pipe 3 switch #: 6

Thus, it will now be understood that there has been disclosed a physically realizable one terabit per second or more ATM packet switch which has a large number of inputs and a large number of outputs. This ATM switch architecture has multiple pipes and uses principles of independence, preferences, rolling and distributed control to keep ATM cell losses very low while achieving such large data throughputs. This ATM switch architecture is flexible and can support variable length packets and/or STM data by adjustments to the distributed control system, thereby allowing evolvable STM/ATM systems.

Claims

1. A packet switch for switching a telecommunication packet from a plurality of input lines to a plurality of output lines, comprising:
a plurality of input interfaces, each having an input port connected to a respective input line of said plurality of input lines, and each of said input interfaces having an output port;
a network for switching a plurality of I network input ports to a plurality of P network output ports;
each of said plurality of input interface output ports is fanned out to a respective group of F of said I network input ports;
said network having a plurality of C pipes, where C is an integer of a value equal to P/I;
a plurality of output modules, said output modules together having a plurality of inputs, each of said output module inputs connected to respective network output port of said plurality of P network output ports, and together having a plurality of outputs, each of said output module outputs connected to a respective output line of said plurality of output lines;
each pipe of said C pipes providing one path for switching a respective telecommunication packet from each of the plurality of inputs lines which is connectable to a respective output line of the plurality of output lines; and
means for hunting a respective path through said packet switch for each telecommunication packet with a low probability of blocking.

2. The packet switch as set forth in claim 1, wherein said path hunting means comprises:
an out of band controller.

3. The packet switch as set forth in claim 2, wherein:
each of said input interfaces has a store for storing a telecommunication packet; and
said out-of-band controller rolls a request for a path for the telecommunications packet which was unable to find an unblocked path through a first pipe to an input of a pipe controller of a second pipe and the telecommunications packet is stored in said input interface while the controller is hunting an unblocked path.

4. The packet switch as set forth in claim 2, wherein:
each of said input interfaces has a store for storing a telecommunication packet; and
said out-of-band controller rolls a request for a path for the telecommunications packet which was unable to

find an unblocked path through both a first pipe and a second pipe to an input of a pipe controller of a third pipe and the telecommunication packet is stored in said input interface while the controller is hunting an unblocked path.

5. The packet switch as set forth in claim 2, wherein:
each of said input interfaces has a store for storing a telecommunication packet; and
said out-of-band controller rolls a request for a path for the telecommunications packet which was unable to find an unblocked path through a first pipe, a second pipe and a third pipe to an input of a pipe controller of a fourth pipe and the telecommunication packet is stored in said input interface while the controller is hunting an unblocked path.

6. The packet switch as set forth in claim 2, wherein said out-of-band controller assigns an order of preference to ATM cells to reduce a probability of losing an ATM cell by internal blocking.

7. The packet switch as set forth in claim 2, wherein said out-of-band controller assigns an order of preference to ATM cells, and rolls a request for a path for a telecommunications packet which was unable to find an unblocked path through a first pipe to an input of a pipe controller of a second pipe.

8. The packet switch as set forth in claim 2, wherein said out-of-band controller assigns an order of preference to ATM cells, and rolls a request for a path for a telecommunications packet which was unable to find an unblocked path through a first pipe and a second pipe to an input of a pipe controller of a third pipe.

9. The packet switch as set forth in claim 2, wherein said out-of-band controller assigns an order of preference to ATM cells, and rolls a request for a path for a telecommunications packet which was unable to find an unblocked path through a first pipe, a second pipe and at third pipe to an input of a pipe controller of a fourth pipe.

10. A packet switch for switching telecommunication packets, comprising:
a network for switching a plurality of I inputs to a plurality of P outputs;
said network having a plurality of C pipes, where C is an integer of a value equal to P/I;
each pipe having a respective pattern of switching its inputs to its outputs;
each pattern of switching of a pipe is independent of the patterns of switching of the other pipes;
a plurality of output modules connected to said plurality of P outputs; and
means for hunting a path through said packet switch for a telecommunication packet.

11. The packet switch as set forth in claim 10, wherein said path hunting means comprises:
an out of band controller.

12. The packet switch as set forth in claim 11, wherein said out-of-band controller rolls a request for a path of a telecommunication packet which was unable to find an unblocked path through a first pipe to an input of a second pipe.

13. The packet switch as set forth in claim 11, wherein said out-of-band controller rolls a request for a path of a telecommunication packet which was unable to find an unblocked path through both a first pipe and a second pipe to an input of a pipe controller of a third pipe.

14. The packet switch as set forth in claim 11, wherein said out-of-band controller rolls a request for a path of a telecommunication packet which was unable to find an unblocked path through a first pipe, a second pipe and a third pipe to an input of a pipe controller of a fourth pipe.

15. The packet switch as set forth in claim 11, wherein said out-of-band controller assigns an order of preference to ATM cells to reduce a probability of losing an ATM cell by internal blocking.

16. The packet switch as set forth in claim 11, wherein said out-of-band controller assigns an order of preference to ATM cells, and rolls a request for a path of a telecommunications packet which was unable to find an unblocked path through a first pipe to an input of a pipe controller of a second pipe.

17. The packet switch as set forth in claim 11, wherein said out-of-band controller assigns an order of preference to ATM cells, and rolls a request for a path of a telecommunication packet which was unable to find an unblocked path through a first pipe and a second pipe to an input of a pipe controller of a third pipe.

18. The packet switch as set forth in claim 11, wherein said out-of-band controller assigns an order of preference to ATM cells, and rolls a request for a path of a telecommunications packets which was unable to find an unblocked path through a first pipe, a second pipe and a third pipe to an input of a fourth pipe.

19. The packet switch as set forth in claim 11, wherein said out-of-band controller controls finds and establishes paths through said packet switch for the telecommunication packets.

20. An ATM switch for switching ATM packets, comprising:
a plurality of ATM interface cards, each having a respective input connected to an ATM telecommunication line and an output;
a network for switching a plurality of I inputs to a plurality of P outputs;
said network having a plurality of F pipes, where F is an integer of a value equal to P/I;
each of said outputs of said input interfaces is fanned out by a factor of F and connected a respective input of each of said F pipes;
each pipe having a respective pattern of switching its inputs to its outputs;
each pattern of switching of a pipe is independent of the patterns of switching of the other pipes;
a plurality of output modules connected to said plurality of P outputs;
each of said output modules having a plurality of outputs; and
means for hunting a path from an input of one of said plurality of input interfaces through said ATM switch for an ATM packet to a desired output of one of said plurality of output modules.

21. The ATM switch according to claim 20, wherein an input interface of said plurality of input interfaces has a memory that stores an ATM packet for two ATM packet periods to allow said means for hunting a path to hunt a path for an ATM packet which was blocked during the previous ATM packet period from an input of one of said plurality of input interfaces through said ATM switch for an ATM packet to a desired output of one of said plurality of output modules during the next ATM packet period.

22. The ATM switch according to claim 21, wherein said ATM packet which was blocked during the previous ATM packet period is communicated through said ATM switch to its respective desired output of its output module in proper sequence without the use of a time stamp.

23. A method for communicating a telecommunication packet through a packet switch having a plurality input interfaces, each of said input interfaces having a store for storing a telecommunications packet, and a plurality of pipes, said method comprising the steps of:

a. storing a telecommunication packet in an input interface;
b. hunting during a first time period a non-blocked path for said telecommunication packet through a first pipe to an output module which is connected to a destination output line;
c. if a non-blocked path is found through said first pipe skipping to step k, otherwise continuing to step d;
d. if a non-blocked path is not found through said first pipe, hunting during a second time period a non-blocked path for said telecommunication packet through a second pipe to the output module which is connected to the destination output line ;
e. if a non-blocked path is found through said second pipe skipping to step k, otherwise continuing to step f;
f. if a non-blocked path is not found through said second pipe in the second time period, hunting in a third time period a non-blocked path for said telecommunication packet through a third pipe to the output module which is connected to the destination output line;
g. if a non-blocked path is found through said third pipe skipping to step k, otherwise continuing to step h;
h. if a non-blocked path is not found through said third pipe in the third time period, hunting in a fourth time period a non-blocked path for said telecommunication packet through a fourth pipe to the output module which is connected to the destination output line;
i. if a non-blocked path is found through said fourth pipe skipping to step k, otherwise continuing to step j;
j. if a non-blocked path is not found through said fourth pipe in the fourth time period, clearing said telecommunication packet from said input interface and losing said telecommunication packet;
k. communicating said telecommunication packet from said input interface to an output module that is connected to the desired output line; and
l. routing said telecommunication packet through said output module to said desired output line.

24. A method for communicating an ATM packet through a packet switch having a plurality input interfaces, each of said input interfaces having a store for storing an ATM packet , and a plurality of pipes, said method comprising the steps of:

a. communicating an ATM packet from an input line to an input interface of said plurality of input interfaces;
b. determining from a header of said ATM packet a destination output line and forwarding said destination output line determination to an out-of-band controller;
c. storing said ATM packet in said input interface;
d. hunting during a first half of a first ATM packet time period a non-blocked path for said telecommunication packet through a first pipe to an output module which is connected to the destination output line;
e. if a non-blocked path is found through said first pipe skipping to step m, otherwise continuing to step f;
f. if a non-blocked path is not found through said first pipe, hunting during a second half of said first ATM packet time period a non-blocked path for said ATM packet through a second pipe to the output module which is connected to the destination output line ;
g. if a non-blocked path is found through said second pipe skipping to step m, otherwise continuing to step h;
h. if a non-blocked path is not found through said second pipe in the second half of said first ATM packet time period, hunting in a first half of a second ATM packet time period a non-blocked path for said ATM packet through a third pipe to the output module which is connected to the destination output line;
i. if a non-blocked path is found through said third pipe skipping to step m, otherwise continuing to step j;
j. if a non-blocked path is not found through said third pipe in the first half of said second ATM packet time period, hunting in a second half of said second ATM packet time period a non-blocked path for said ATM packet through a fourth pipe to the output module which is connected to the destination output line;
k. if a non-blocked path is found through said fourth pipe skipping to step m, otherwise continuing to step l;
l. if a non-blocked path is not found through said fourth pipe in the fourth time period, clearing said ATM packet from said input interface and losing said ATM packet;
m. communicating said ATM packet from said store of said input interface to said output module that is connected to said destination output line; and
n. routing said ATM packet through said output module to said destination output line.

25. A method for communicating an ATM packet through a packet switch having a plurality input interfaces, each of said input interfaces having a store for storing an ATM packet , and a plurality of pipes, said method comprising the steps of:

a. communicating an ATM packet from an input line to an input interface of said plurality of input interfaces;
b. determining from a header of said ATM packet a destination output line and forwarding said destination output line determination to an out-of-band controller;
c. storing said ATM packet in said input interface;
d. hunting during a first half of a first ATM packet time period a non-blocked path for said telecommunication packet through a first pipe to an output module which is connected to the destination output line;
e. if a non-blocked path is found through said first pipe skipping to step m, otherwise continuing to step f;
f. if a non-blocked path is not found through said first pipe, hunting during a second half of said first ATM packet time period a non-blocked path for said ATM packet through a second pipe to the output module which is connected to the destination output line ;
g. if a non-blocked path is found through said second pipe skipping to step m, otherwise continuing to step h;
h. if a non-blocked path is not found through said second pipe in the second half of said first ATM packet time period, hunting in a first half of a second ATM packet time period a non-blocked path for said ATM packet through a third pipe to the output module which is connected to the destination output line;
i. if a non-blocked path is found through said third pipe skipping to step m, otherwise continuing to step j;
j. if a non-blocked path is not found through said third pipe in the first half of said second ATM packet time period, hunting in a second half of said second ATM packet time period a non-blocked path for said ATM packet through a fourth pipe to the output module which is connected to the destination output line;
k. if a non-blocked path is found through said fourth pipe skipping to step m, otherwise continuing to step l;
l. if a non-blocked path is not found through said fourth pipe in the fourth time period, clearing said ATM packet from said input interface and losing said ATM packet;
m. communicating said ATM packet from said store of said input interface to said output module that is connected to said destination output line; and
n. routing said ATM packet through said output module to its destination output line such that said ATM packet is in an appropriate time sequence with respect to any other ATM packet that received by any of said plurality of input interfaces during said first and second ATM packet time periods.

26. A method for communicating a plurality of ATM packets from a plurality of input lines through a packet switch having a plurality input interfaces, each of said input interfaces having an input connected to a respective input line of said plurality of input lines and a store for storing a respective ATM packet of said plurality of ATM packets, and a plurality of pipes, said method comprising the steps of:

a. communicating each ATM packet from an input line to a respective input interface of said plurality of input interfaces;
b. determining from header data of each ATM packet a respective destination output line;
c. assigning each ATM packet a first phase preference level that is different from another first phase preference level;
d. communicating said first phase preference level and said destination output line determination to an out-of-band controller;
e. storing each ATM packet in its input interface until it is communicated through one of said plurality of pipes or cleared;
f. hunting during a first half of a first ATM packet time period a non-blocked path for said telecommunication packet through a first pipe to an output module which is connected to the destination output line;
e. if a non-blocked path is found through said first pipe skipping to step m, otherwise continuing to step f;
f. if a non-blocked path is not found through said first pipe, hunting during a second half of said first ATM packet line period a non-blocked path for said ATM packet through a second pipe to the output module which is connected to the destination output line ;
g. if a non-blocked path is found through said second pipe skipping to step m, otherwise continuing to step h;
h. if a non-blocked path is not found through said second pipe in the second half of said first ATM packet time period, hunting in a first half of a second ATM packet time period a non-blocked path for said ATM packet through a third pipe to the output module which is connected to the destination output line;
i. if a non-blocked path is found through said third pipe skipping to step m, otherwise continuing to step j;
j. if a non-blocked path is not found through said third pipe in the first half of said second ATM packet time period, hunting in a second half of said second ATM packet time period a non-blocked path for said ATM packet through a fourth pipe to the output module which is connected to the destination output line;
k. if a non-blocked path is found through said fourth pipe skipping to step m, otherwise continuing to step l;
l. if a non-blocked path is not found through said fourth pipe in the fourth time period, clearing said ATM packet from said input interface and losing said ATM packet;
m. communicating said ATM packet from said store of said input interface to said output module that is connected to said destination output line; and
n. routing said ATM packet through said output module to said destination output line.

27. A controller for a packet switch for locating a path for each telecommunication packet, comprising:
memory means for storing a busy-or-idle status for each connection between a single stage switch fabric and a plurality of output modules;
said memory means is divided into a plurality of be-or-idle status tables; and
means for accessing each of said plurality of busy-or-idle status tables concurrently for locating the path for each telecommunication packet to the desired output module of said plurality of output modules.

28. A controller for a packet switch having a switch fabric having a plurality of pipes with each pipe having a plurality of crossbar switches for locating a path for each telecommunication packet, comprising:
a plurality of pipe hunting controllers, with each of said pipe hunting controllers controlling a respective pipe of said plurality of pipes;
each of said pipe hunting controllers having a plurality of crossbar switch controllers, with each crossbar switch of a pipe having a respective crossbar switch controller;
a plurality of storage means for storing busy-or-idle tables;
each of said plurality of storage means is associated with a respective crossbar switch controller of said plurality of crossbar switch controllers and stores a respective busy-or-idle status table therefor; and
means for accessing each of said busy-or-idle status tables concurrently.

29. The controller for a packet switch as set forth in claim 28, wherein each of said busy-or-idle tables has a plurality of busy-or-idle status bits with each busy-or-idle status bit corresponding to a busy-or-idle status of a respective output of its respective crossbar switch.

30. The controller for a packet switch as set forth in claim 29, wherein each of said crossbar switch controllers processes its plurality of busy-or-idle status bits concurrently.

31. The controller for a packet switch as set forth in claim 29, wherein each busy-or-idle status bit is stored in a respective output controller corresponding to the respective output of its respective crossbar switch.

32. The controller for a packet switch as set forth in claim 31, wherein each output controller processes one busy-or-idle bit for controlling its respective output for each packet processing time.

33. A controller for a packet switch having a switch fabric having a lurality of pipes with each pipe having a plurality of crossbar switches for locating a path for each telecommunication packet, comprising:

a plurality of pipe hunting controllers, with each of said pipe hunting controllers controlling a respective pipe of said plurality of pipes;

each of said pipe hunting controllers having a plurality of crossbar switch controllers, with each crossbar switch of a pipe being controlled by its respective crossbar switch controller;

a plurality of storage means for storing busy-or-idle tables;

each of said plurality of storage means is associated with a respective crossbar switch controller of said plurality of crossbar switch controllers and stores a respective busy-or-idle table therefor; and

means for accessing each of said busy-or-idle status tables concurrently; wherein each of said storage means comprises:

means for storing a busy-or-idle bit;

means for reading said busy-or-idle bit from said storage means;

means for logically operating on said busy-or-idle bit with a request bit to produce a resulting busy-or-idle bit that is stored in said storage means, a request-out bit and a connect bit; and

said connect bit cause a path to be connected through its respective crossbar switch if said connect bit is set.

34. The controller for a packet switch as set forth in claim 33, further comprising:

means for forwarding said request-out bit to another of said plurality of storage means in another pipe to hunt a path where said request-out bit is inputted to a respective logical operating means of said another storage means and logically operated on with another busy-or-idle bit to produce a resulting busy-or-idle bit, another request-out bit and another connect bit.

35. The controller for a packet switch as set forth in claim 33, wherein sixteen request bits are transmitted as a request vector to sixteen storage means in parallel.

36. A method for controlling a packet switch having a switch fabric having a plurality of pipes with each pipe having a plurality of NxN crossbar switches, and a plurality of output modules during each packet cycle period, comprising the steps of:

setting all busy-or-idle bits to idle at a beginning of each packet cycle period;

reading N busy-or-idle bits from a busy-or-idle memory of a crossbar switch controller in parallel as a busy-or-idle vector;

performing parallel pipe hunting operations by logically combining said busy-or-idle vector and a first input request vector to produce a first updated busy-or-idle vector, a connect vector and an output request vector;

storing said updated busy-or-idle vector into the busy-or-idle memory in parallel; and

forwarding said connect vector to a first crossbar switch of said plurality of crossbar switches to set a path from an input of said switch fabric to an output of said switch fabric for a packet.

37. The method of claim 36 wherein the size of each of the NxN crossbar switches is 16X16.

38. The method of claim 36 or 37, further comprising the step of:

forwarding said output request vector which contains any unfulfilled requests from said input request vector to a second crossbar switch controller in another pipe;

using said output request vector as a second input request vector for a second crossbar switch of said plurality of crossbar switches connected to and controlled by said second crossbar switch controller;

reading N busy-or-idle bits from a busy-or-idle memory of said second crossbar switch controller in parallel;

performing parallel pipe hunting operations logically combining said busy-or-idle vector and said input request vector producing a second updated busy-or-idle vector, a second connect vector and a second output request vector;

storing said second updated busy-or-idle vector into the busy-or-idle memory of said second crossbar switch in parallel; and

forwarding said second connect vector to said second crossbar switch to set a path from an input of said switch fabric to an output of said switch fabric for a packet.

39. The method of claim 36 or 37, wherein said performing parallel pipe hunting operations step includes the step of concurrently processing N busy-idle bits by providing N link controllers within each switch controller with each link controller processing busy-idle bits for a respective intermediate link between the switch fabric and an output module of said plurality output modules.

40. A network for switching a plurality of I inputs to a plurality of P outputs, comprising;
a plurality of C pipes, where C is an integer of a value equal to P/I;
each pipe having a respective pattern of connections from its inputs to its outputs;
each pattern of connections is independent of any other pattern of connections; and
said independence of the patterns causes reductions in the internal blocking rate of network.

41. The apparatus as set forth in claim 10 or 40, wherein the independence of each of said patterns of connections is determined according to Galois field theory.

42. For use in a telecommunication switch having N input lines and N output lines, a switch network, which has N inputs, P outputs and a fan-out of F with F being equal to a ratio of P/N, comprising:
a plurality of crossbar switches of number X, each having I/X inputs and P/FX outputs arranged in a single stage, where X/F is an integer that is greater than 1;
each of said N input lines fanning out to a respective F input of said I input;
each of said P outputs is connected to a respective input of an output module of a plurality of output modules,
each of said output modules is a buffered concentrator concentrating F inputs into a respective output line; and
the number of inputs I/X of each crossbar switch of said plurality of crossbar switches is less than the product of a number of input lines N and the value of the fan-in F.

43. The switch network according to claim 42, wherein
the number of inputs I is an integer multiple of the number of input lines N.

44. The switch network according to claim 42, wherein
each crossbar switch of said plurality of crossbar switches is a 16x16 crossbar switch.

45. The switch network according to claim 42, wherein said plurality of crossbar switches are divided into F pipes with each pipe having a respective connection that is connectable from an input via one of said plurality of output modules to each of said N output lines.

46. The switch network according to claim 45, wherein each of said crossbar switches is an electronic device.

47. The switch network according to claim 45, wherein each of said switches is a photonic device.

48. A single stage switch fabric partitioned into multiple partitions, said switch fabric having inputs to each partition produced according to a method comprising the steps of:
representing each input port (I) by an eight-bit binary number (i_7, i_6, i_5, i_4, i_3, i_2, i_1, I_0) which is designated S;
representing each 16x16 crossbar switch S_θ (I) to which the input port (I) is connected in partition θ by a six-bit binary number (s_5, s_4, s_3, s_2, s_1, s_0) θ ; and
connecting each input port (I) to an input on a respective crossbar switch S_θ (I) in each of said partitions according to Galois field theory to effectively reduce the ATM cell blocking probability thereof.

49. The single stage switch fabric as set forth in claim 48, wherein the mapping functions for determining the connections according to Galois field theory are:
(s_5, s_4, s_3, s_2, s_1, s_0) 0=(0, 0, i_3, i_2, i_1 i_0);
(s_5, s_4, s_3, s_2, s_1, s_0) 1=(0 ,1, i_7 XOR i_3, i_6 XOR i_2, i_5, XOR i_1, i_4 XOR i_0);
(s_5, s_4, s_3, s_2, s_1, s_0) 2=(1, 0, i_7 XOR i_2, i_6 XOR i_1, i_5 XOR i_0 XOR i_7, i_4 XOR i_7) and
(s_5, s_4, s_3, s_2, s_1) 3=(1, 1, i_7 XOR i_1, i_6 XOR i_0 XOR i_7, i_5 XOR i_7 XOR i_2, i_4 XOR I_2).

50. A switch fabric for use in a telecommunication having a plurality of input lines connected to a plurality of line interfaces and a plurality of output lines connected to outputs of a plurality of output modules, said switch fabric comprising:
a plurality of pipes with each pipe having a plurality of inputs, with each of said plurality of pipe inputs connecting to a respective output of said plurality of line interfaces such that each of said line interfaces has a fan-out equal to the number of said plurality of pipes;

each of said pipes is a single stage network and provides a single path from any line interface output to an output line via an output module of the plurality of output modules;

each of said output modules having a concentration ratio that is equal in number to the amount of the fan-out of the line interface outputs; and

each path between an input line and its desired output line is sufficiently independent from the other paths through the other pipes between that input line and that desired output line that the probability of internal block is low

wherein reducing the number of paths available to connect any input line to any output line to the same as the number of pipes substantially reduces the number of active devices needed for connections without an equivalent rise in the internal blocking probability.

51. A packet switch for switching circuit switched communications from a plurality of circuit switched input lines to a plurality of circuit switched output lines and packet switched communications from a plurality of packet switched input lines to a plurality of packet switched output lines, comprising:

a plurality of circuit switched input interfaces, each having an input port connected to a respective circuit switched input line of said plurality of circuit switched input lines, and each of said circuit switched input interfaces having an output port;

a plurality of packet switched input interfaces, each having an input port connected to a respective packet switched input line of said plurality of packet switched input lines, and each of said packet switched input interfaces having an output port;

a single stage switch fabric having a plurality of input ports with a first portion of said input ports connected to respective output ports of said circuit switched input interfaces and a second portion of said input ports connected to respective output ports of said packet switched input interfaces;

a plurality of circuit switched output modules, said circuit switched output modules together having a plurality of inputs, each of said circuit switched output module inputs connected to respective output port of said first portion of said single stage switching fabric, and together having a plurality of outputs, each of said circuit switched output module outputs connected to a respective circuit switched output line of said plurality of circuit switched output lines;

a plurality of packet switched output modules, said packet switched output modules together having a plurality of inputs, each of said packet switched output module inputs connected to respective output port of said second portion of said single stage switching fabric, and together having a plurality of outputs, each of said packet switched output module outputs connected to a respective packet switched output line of said plurality of packet switched output lines;

means for hunting a path through said switch fabric to a desired circuit switched output line for communication on each circuit switched input line; and

means for hunting a path through said switch fabric to a desired packet switched output line for a packet on each packet switched input line.

52. The switch as set forth in claim 51, wherein:

said circuit switched communication path hunting means includes a first out of band, controller;

said packet switched communication path hunting means includes a second out of band, controller;

said switch fabric is partitioned into multiple pipes; and

said out-of-band controllers roll requests which have been denied a path through a first pipe to a second pipe.

53. The switch as set forth in claim 52, wherein said out-of-band controllers roll requests which have been denied a path through a first pipe and a second pipe to a third pipe.

54. The switch as set forth in claim 52, wherein said out-of-band controllers roll requests which have been denied a path through a first pipe, a second pipe and a third pipe to a fourth pipe.

55. The switch as set forth in claim 52, wherein said second out-of-band controller assigns an order of preference to communication packets.

56. The switch as set forth in claim 51, wherein a size of said first portion of said switch fabric is in a range of zero to one hundred per cent.

57. The switch as set forth in claim 56, wherein a length of each communication packet may vary from one packet to another.

58. The switch as set forth in claim 51, 52, 53, 54, 55, 56 or 57, wherein a length of each communication packet may vary from one packet to another.

59. A packet switch as set forth in claims 1 or 10, further comprising
a spare pipe for on-line replacement of any one of said C pipes which is faulty.

60. A growable packet switch for switching packets of data from a plurality of input lines to a plurality of output lines, comprising:
a plurality of input interfaces, each having an input port connected to a respective input line of said plurality of input lines, and an output port;
a single state distribution network for switching a plurality of I input ports to a plurality of L output ports;
each of said input interfaces having a store therein for holding a packet of data until the packet of data is routed to the network;
each of said plurality of input interface output ports fanning out to a respective group of F of said I input ports of said single stage distribution network;
said single stage distribution network having a plurality of P partitions, where P is an integer;
a plurality of output modules, said output modules together having a plurality of inputs, each of said output module inputs connected to respective output port of said plurality of L output ports, and together having a plurality of outputs, each of said output module outputs connected to a respective output line of said plurality of output lines;
an out-of-band controller having a plurality of busy-idle memory units equal in number to the number of partitions P, each of said busy-idle memory units controls the routing of data packets through its respective partition; and
each partition of said P partitions having at least one path from each of the plurality of inputs lines that is connectable to a respective output line of the plurality of output lines.

61. The growable packet switch for switching packets of data from a plurality of input lines to a plurality of output lines according to claim 60,
wherein each data packet routing request that is made in an i-th time interval is completed in either the i-th time interval or in the next subsequent time interval.

62. The growable packet switch for switching packets of data from a plurality of input lines to a plurality of output lines according to claim 61, wherein each of said output modules takes data from its FIFOs such that a cell communicated via an input line in the i-th interval that was routed in the next interval after the i-th interval will be output to the desired output line before any other cell that is communicated via said input line in any interval after the i-th interval.

63. The growable packet switch for switching packets of data from a plurality of input lines to a plurality of output lines according to claim 61, wherein said single stage distribution network is a single electronic crossbar switch.

64. The growable packet switch for switching packets of data from a plurality of input lines to a plurality of output lines according to claim 61, wherein said single stage distribution network is an isomorph of a single electronic crossbar switch which has a plurality of partitions and each partition is a single electronic crossbar switch.

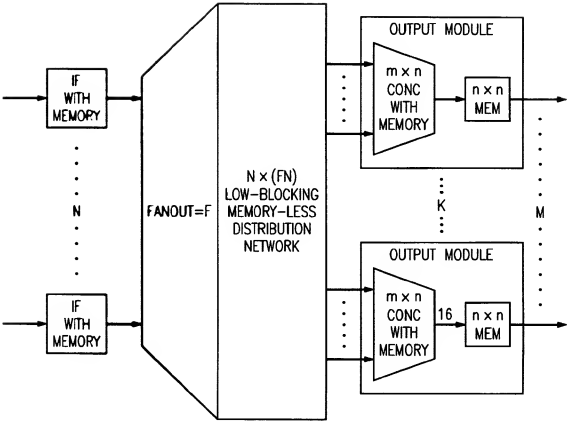
65. The growable packet switch for switching packets of data from a plurality of input lines to a plurality of output lines according to claim 61, wherein said single stage distribution network has a plurality of partitions and each partition has a plurality of electronic crossbar switches, and each partition has at least one path connectable between each input line and each output line.

66. The growable packet switch for switching packets of data from a plurality of input lines to a plurality of output lines according to claim 61, wherein any routing request that arrived during the i-th interval attempts to locate an available path in the i-th interval or in the next subsequent interval to the i-th interval.

67. A growable packet switch for switching packets of data from a plurality of input lines to a plurality of output lines according to claim 61, wherein any routing request that arrived during the i-th interval and passes from busy-idle memory (P-1) to busy-idle memory 0 attempts to locate available paths through the remaining partitions during the next interval after the i-th interval.

68. A method for controlling paths for X ATM cells entering a growable packet switch during an i-th arrival interval that are destined to a common mxn output packet module with X>m, comprising the steps of:

routing m of said X ATM cells during an i-th routing interval; and
rolling the remaining X-m of said X ATM cells into a next routing interval for routing to the output packet module.



(PRIOR ART)

FIG. 1

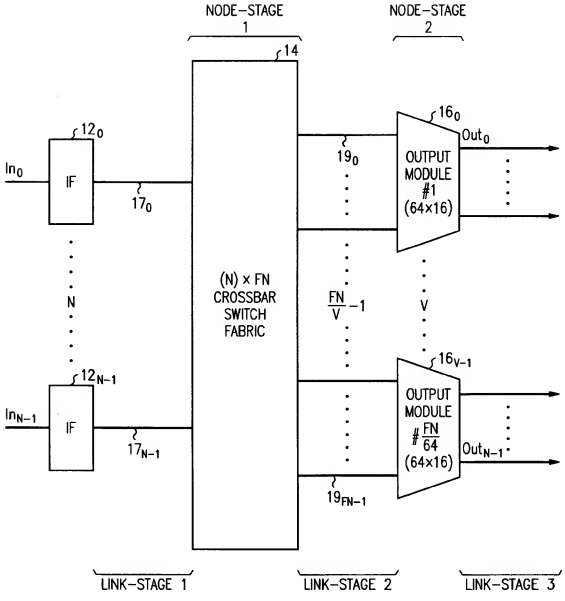


FIG. 2

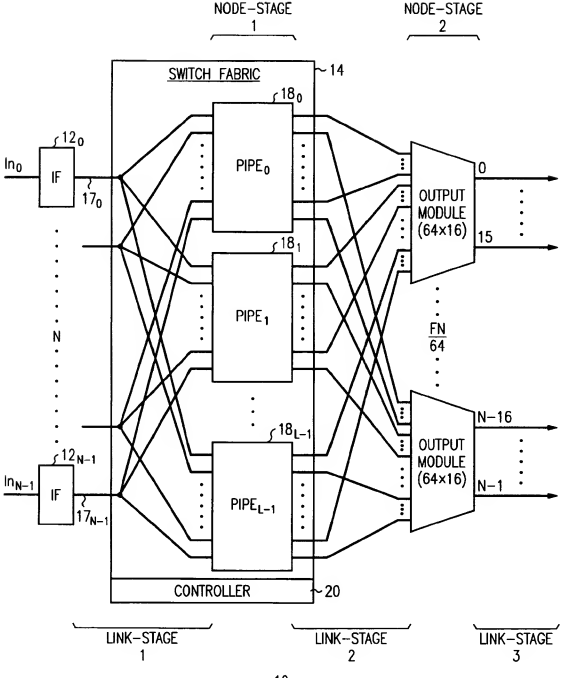

NODE–STAGE 1
NODE–STAGE 2
SWITCH FABRIC
14
18_0
12_0
In_0
IF
17_0
$PIPE_0$
OUTPUT MODULE (64x16)
0
15
18_1
$PIPE_1$
N
FN 64
18_{L-1}
12_{N-1}
In_{N-1}
IF
17_{N-1}
$PIPE_{L-1}$
OUTPUT MODULE (64x16)
N-16
N-1
CONTROLLER
20
LINK–STAGE 1
LINK–STAGE 2
LINK–STAGE 3
10

FIG. 3

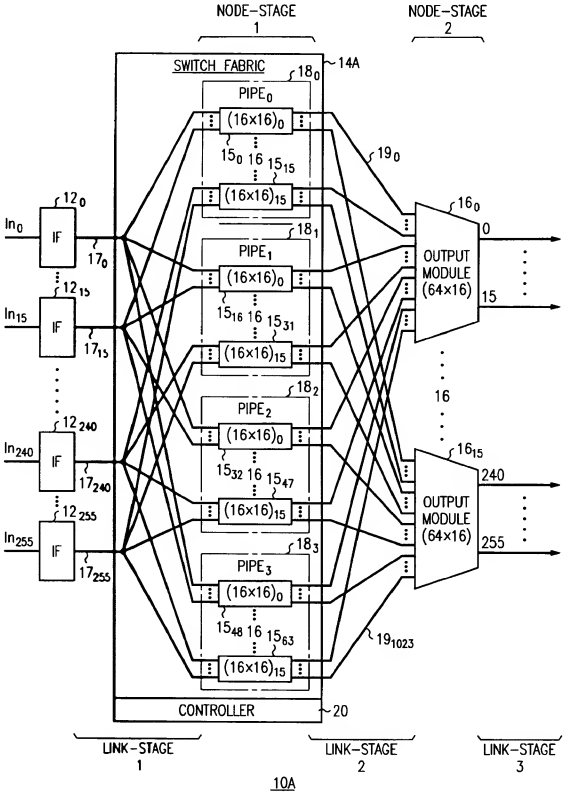


FIG. 4

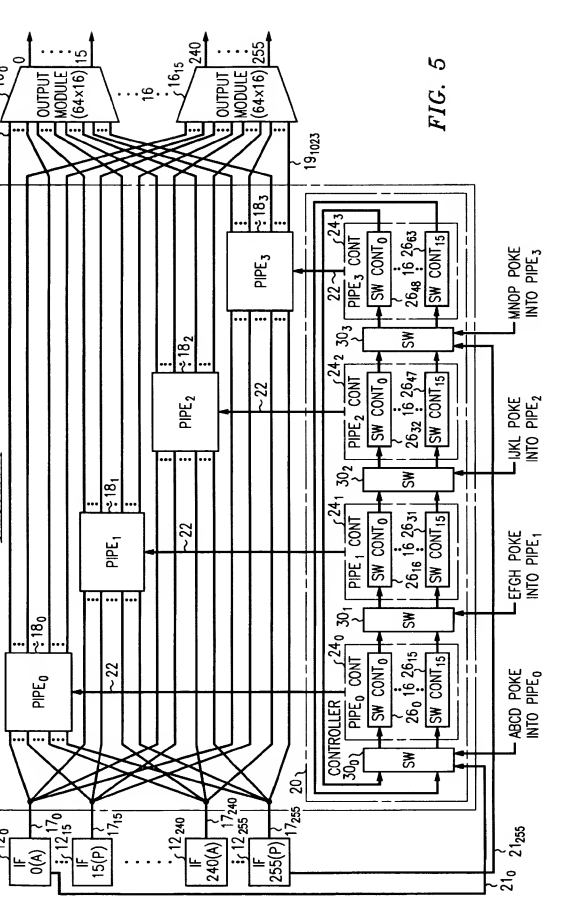

SWITCH FABRIC
14A
PIPE0
PIPE1
PIPE2
PIPE3
22
CONTROLLER
SW
SW CONT0
SW CONT15
PIPE0 CONT
PIPE1 CONT
PIPE2 CONT
PIPE3 CONT
OUTPUT MODULE (64x16)
IF 0(A)
IF 15(P)
IF 240(A)
IF 255(P)
ABCD POKE INTO PIPE0
EFGH POKE INTO PIPE1
IJKL POKE INTO PIPE2
MNOP POKE INTO PIPE3

FIG. 5

TIME	$PIPE_0$ CONTROLLER	$PIPE_1$ CONTROLLER	$PIPE_2$ CONTROLLER	$PIPE_3$ CONTROLLER
i, 0	GROUP 1, PERIOD i			
i, 1		GROUP 1, PERIOD i		
i, 2		GROUP 2, PERIOD i	GROUP 1, PERIOD i	
i, 3			GROUP 2, PERIOD i	GROUP 1, PERIOD i
i, 4			GROUP 3, PERIOD i	GROUP 2, PERIOD i
i, 5	GROUP 2, PERIOD i+1			GROUP 3, PERIOD i
i, 6	GROUP 3, PERIOD i+1			GROUP 4, PERIOD i
i, 7	GROUP 4, PERIOD i+1	GROUP 3, PERIOD i+1		
i, 8		GROUP 4, PERIOD i+1		
i, 9			GROUP 4, PERIOD i+1	

FIG. 6

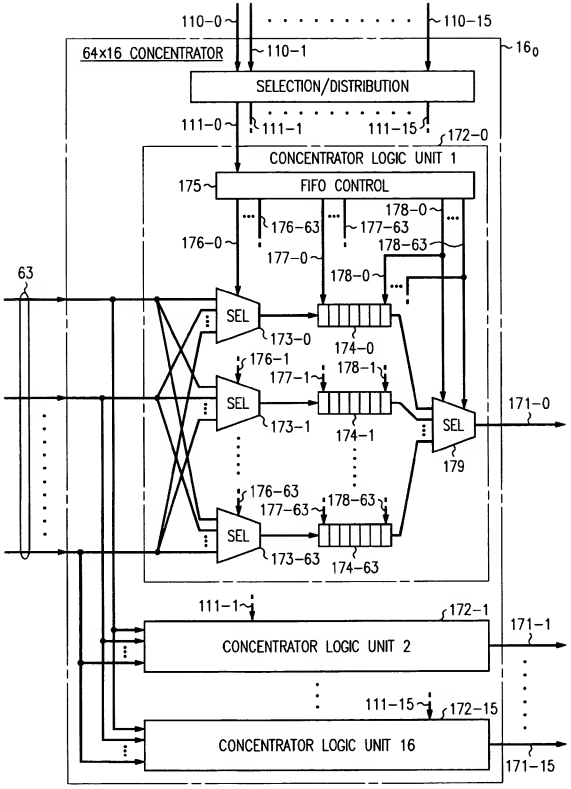

110-0
110-15
64x16 CONCENTRATOR
110-1
16₀
SELECTION/DISTRIBUTION
111-0
111-1
111-15
172-0
CONCENTRATOR LOGIC UNIT 1
175
FIFO CONTROL
178-0
176-63
177-63
176-0
178-63
177-0
178-0
63
SEL
173-0
174-0
176-1
177-1
178-1
171-0
173-1
174-1
179
176-63
177-63
178-63
173-63
174-63
111-1
172-1
171-1
CONCENTRATOR LOGIC UNIT 2
111-15
172-15
CONCENTRATOR LOGIC UNIT 16
171-15

FIG. 7

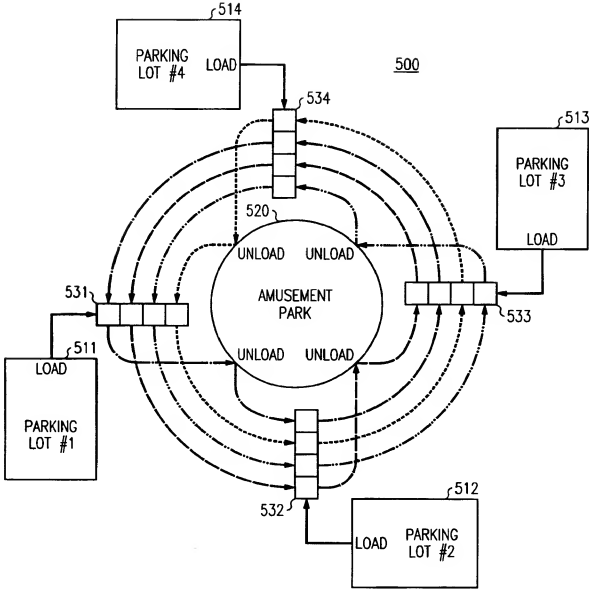


FIG. 8

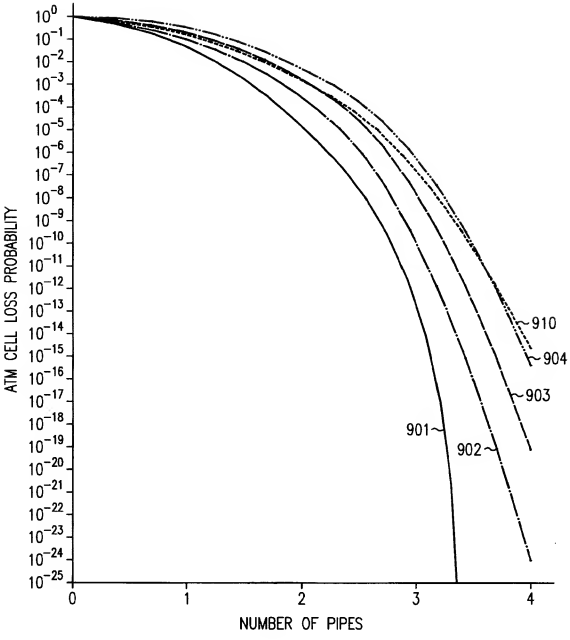

ATM CELL LOSS PROBABILITY
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
10^{-9}
10^{-10}
10^{-11}
10^{-12}
10^{-13}
10^{-14}
10^{-15}
10^{-16}
10^{-17}
10^{-18}
10^{-19}
10^{-20}
10^{-21}
10^{-22}
10^{-23}
10^{-24}
10^{-25}
0
1
2
3
4
NUMBER OF PIPES
910
904
903
901
902

FIG. 9

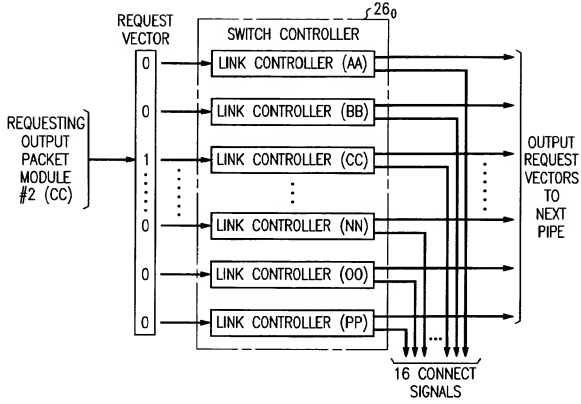


FIG. 10

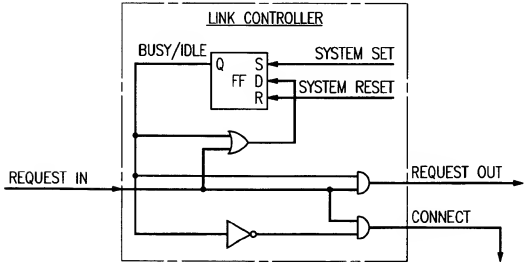


FIG. 11

FIG. 12

REQUEST IN	CURRENT BUSY/IDLE	REQUEST OUT	CONNECT	NEXT BUSY/IDLE
0	0	0	0	0
0	1	0	0	1
1	0	0	1	1
1	1	1	0	1

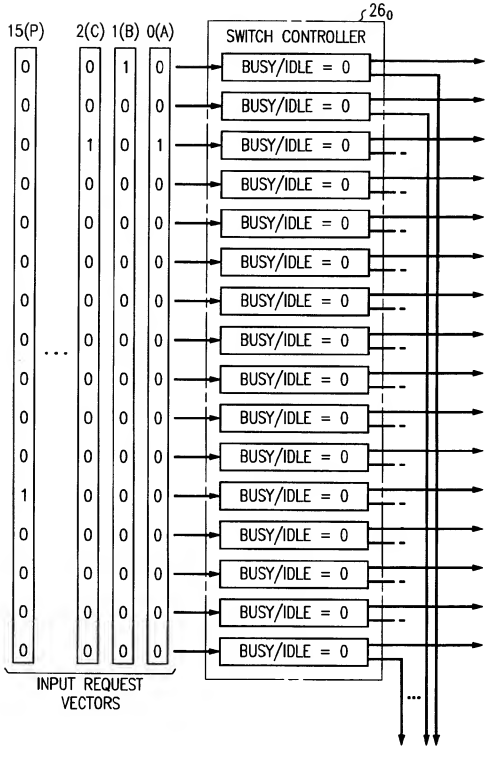


FIG. 13A

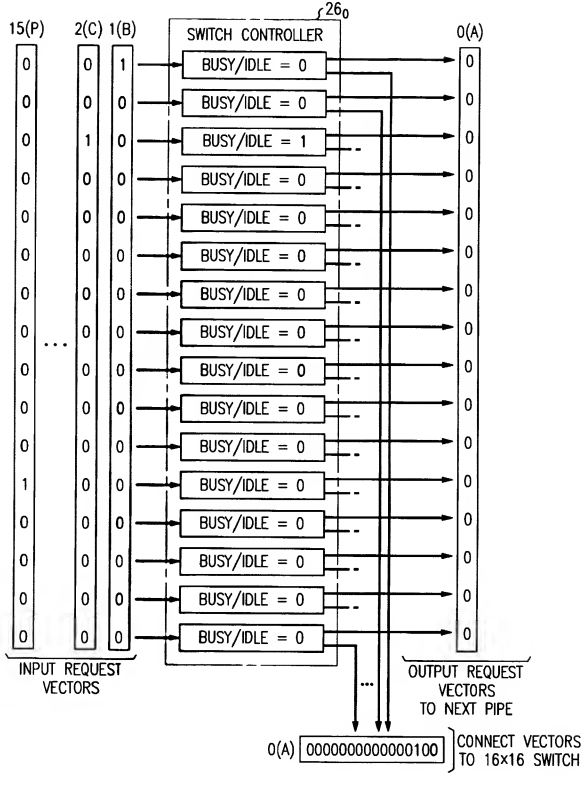

26₀
15(P)
2(C)
1(B)
SWITCH CONTROLLER
0(A)
BUSY/IDLE = 0
BUSY/IDLE = 0
BUSY/IDLE = 1
BUSY/IDLE = 0
BUSY/IDLE = 0
BUSY/IDLE = 0
BUSY/IDLE = 0
BUSY/IDLE = 0
BUSY/IDLE = 0
BUSY/IDLE = 0
BUSY/IDLE = 0
BUSY/IDLE = 0
BUSY/IDLE = 0
BUSY/IDLE = 0
BUSY/IDLE = 0
BUSY/IDLE = 0
INPUT REQUEST VECTORS
OUTPUT REQUEST VECTORS TO NEXT PIPE
0(A) 0000000000000100
CONNECT VECTORS TO 16x16 SWITCH

FIG. 13B

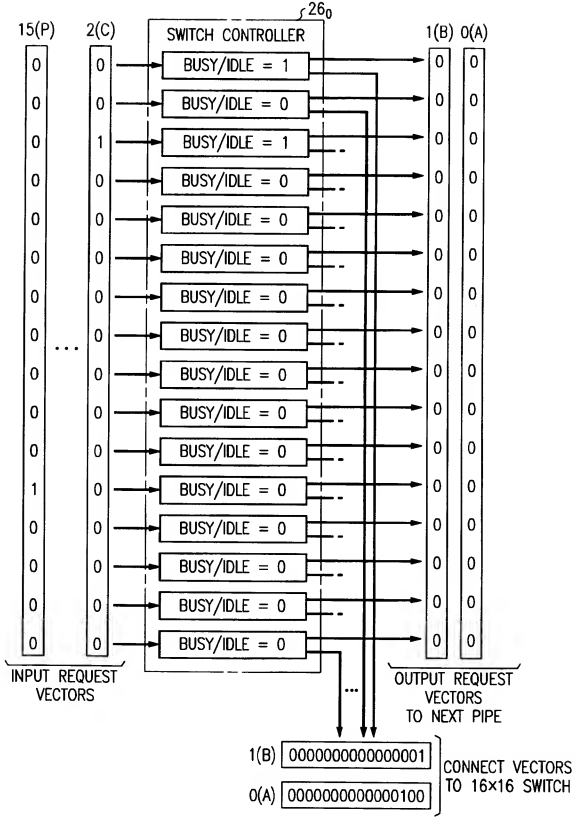


FIG. 13C

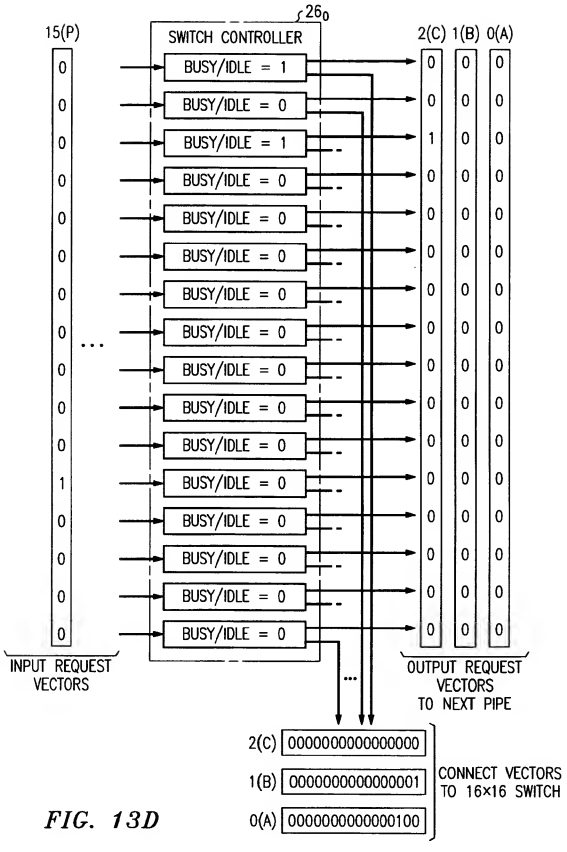

26₀
15(P)
SWITCH CONTROLLER
2(C) 1(B) 0(A)
BUSY/IDLE = 1
BUSY/IDLE = 0
BUSY/IDLE = 1
BUSY/IDLE = 0
BUSY/IDLE = 0
BUSY/IDLE = 0
BUSY/IDLE = 0
BUSY/IDLE = 0
BUSY/IDLE = 0
BUSY/IDLE = 0
BUSY/IDLE = 0
BUSY/IDLE = 0
BUSY/IDLE = 0
BUSY/IDLE = 0
BUSY/IDLE = 0
BUSY/IDLE = 0
INPUT REQUEST VECTORS
OUTPUT REQUEST VECTORS TO NEXT PIPE
2(C) 0000000000000000
1(B) 0000000000000001
0(A) 0000000000000100
CONNECT VECTORS TO 16×16 SWITCH

FIG. 13D

FIG. 14

TIME	CONTROLLER 24_0	CONTROLLER 24_1	CONTROLLER 24_2	CONTROLLER 24_3
i−1,17	$R_{N,i-1} \times B_{0,i}$	$R_{M,i-1} \times B_{1,i}$	* 2	$R_{0,i-1} \times B_{3,i-1}$
i−1,18	$R_{0,i-1} \times B_{0,i}$	$R_{N,i-1} \times B_{1,i}$	$R_{M,i-1} \times B_{2,i}$	$R_{P,i-1} \times B_{3,i-1}$
i−1,19	$R_{P,i-1} \times B_{0,i}$	$R_{0,i-1} \times B_{1,i}$	$R_{N,i-1} \times B_{2,i}$	
i,0	$R_{A,i} \times B_{0,i}$	$R_{P,i-1} \times B_{1,i}$	$R_{0,i-1} \times B_{2,i}$	
i,1	$R_{B,i} \times B_{0,i}$	$R_{A,i} \times B_{1,i}$	$R_{P,i-1} \times B_{2,i}$	
i,2	$R_{C,i} \times B_{0,i}$	$R_{B,i} \times B_{1,i}$	$R_{A,i} \times B_{2,i}$	* 3
i,3	$R_{D,i} \times B_{0,i}$	$R_{C,i} \times B_{1,i}$	$R_{B,i} \times B_{2,i}$	$R_{A,i} \times B_{3,i}$
i,4		$R_{D,i} \times B_{1,i}$	$R_{C,i} \times B_{2,i}$	$R_{B,i} \times B_{3,i}$
i,5		$R_{E,i} \times B_{1,i}$	$R_{D,i} \times B_{2,i}$	$R_{C,i} \times B_{3,i}$
i,6		$R_{F,i} \times B_{1,i}$	$R_{E,i} \times B_{2,i}$	$R_{D,i} \times B_{3,i}$
i,7	* 0	$R_{G,i} \times B_{1,i}$	$R_{F,i} \times B_{2,i}$	$R_{E,i} \times B_{3,i}$
i,8	$R_{E,i} \times B_{0,i+1}$	$R_{H,i} \times B_{1,i}$	$R_{G,i} \times B_{2,i}$	$R_{F,i} \times B_{3,i}$
i,9	$R_{F,i} \times B_{0,i+1}$		$R_{H,i} \times B_{2,i}$	$R_{G,i} \times B_{3,i}$
i,10	$R_{G,i} \times B_{0,i+1}$		$R_{I,i} \times B_{2,i}$	$R_{H,i} \times B_{3,i}$
i,11	$R_{H,i} \times B_{0,i+1}$		$R_{J,i} \times B_{2,i}$	$R_{I,i} \times B_{3,i}$
i,12	$R_{I,i} \times B_{0,i+1}$	* 1	$R_{K,i} \times B_{2,i}$	$R_{J,i} \times B_{3,i}$
i,13	$R_{J,i} \times B_{0,i+1}$	$R_{I,i} \times B_{1,i+1}$	$R_{L,i} \times B_{2,i}$	$R_{K,i} \times B_{3,i}$
i,14	$R_{K,i} \times B_{0,i+1}$	$R_{J,i} \times B_{1,i+1}$		$R_{L,i} \times B_{3,i}$
i,15	$R_{L,i} \times B_{0,i+1}$	$R_{K,i} \times B_{1,i+1}$		$R_{M,i} \times B_{3,i}$
i,16	$R_{M,i} \times B_{0,i+1}$	$R_{L,i} \times B_{1,i+1}$		$R_{N,i} \times B_{3,i}$
i,17	$R_{N,i} \times B_{0,i+1}$	$R_{M,i} \times B_{1,i+1}$	* 2	$R_{0,i} \times B_{3,i}$
i,18	$R_{0,i} \times B_{0,i+1}$	$R_{N,i} \times B_{1,i+1}$	$R_{M,i} \times B_{2,i+1}$	$R_{P,i} \times B_{3,i}$
i,19	$R_{P,i} \times B_{0,i+1}$	$R_{0,i} \times B_{1,i+1}$	$R_{N,i} \times B_{2,i+1}$	
i+1,0	$R_{A,i+1} \times B_{0,i+1}$	$R_{P,i} \times B_{1,i+1}$	$R_{0,i} \times B_{2,i+1}$	
i+1,1	$R_{B,i+1} \times B_{0,i+1}$	$R_{A,i+1} \times B_{1,i+1}$	$R_{P,i} \times B_{2,i+1}$	
i+1,2	$R_{C,i+1} \times B_{0,i+1}$	$R_{B,i+1} \times B_{1,i+1}$	$R_{A,i+1} \times B_{2,i+1}$	* 3
i+1,3	$R_{D,i+1} \times B_{0,i+1}$	$R_{C,i+1} \times B_{1,i+1}$	$R_{B,i+1} \times B_{2,i+1}$	$R_{A,i+1} \times B_{3,i+1}$
i+1,4		$R_{D,i+1} \times B_{1,i+1}$	$R_{C,i+1} \times B_{2,i+1}$	$R_{B,i+1} \times B_{3,i+1}$

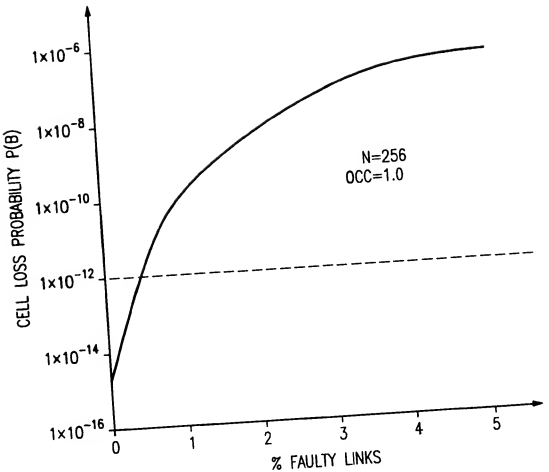

CELL LOSS PROBABILITY P(B)
1x10^-6
1x10^-8
1x10^-10
1x10^-12
1x10^-14
1x10^-16
N=256
OCC=1.0
0
1
2
3
4
5
% FAULTY LINKS

FIG. 15

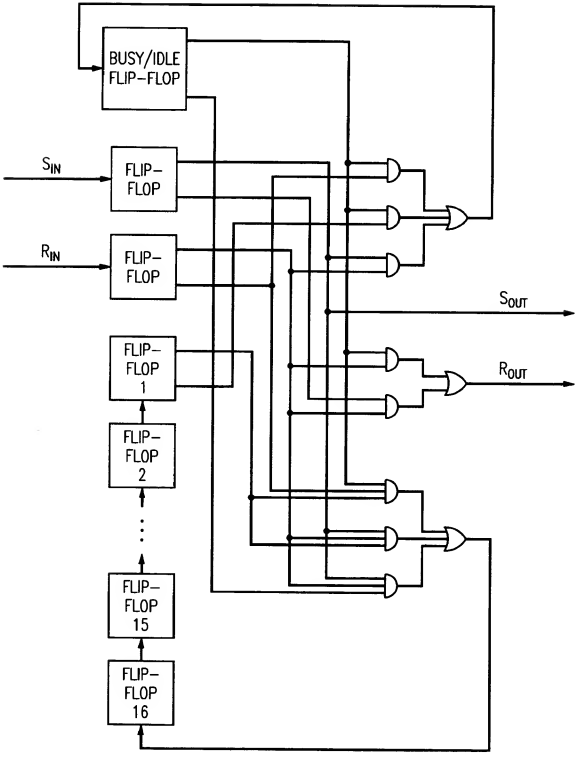


FIG. 16

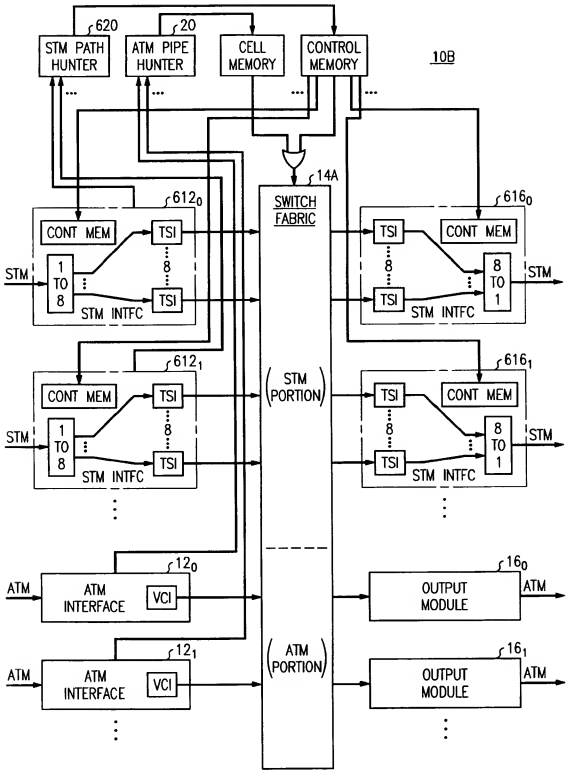

620
20
STM PATH HUNTER
ATM PIPE HUNTER
CELL MEMORY
CONTROL MEMORY
10B
14A
SWITCH FABRIC
612_0
612_1
616_0
616_1
CONT MEM
TSI
1 TO 8
8 TO 1
STM INTFC
STM
STM PORTION
ATM PORTION
12_0
12_1
ATM INTERFACE
VCI
ATM
16_0
16_1
OUTPUT MODULE

FIG. 17

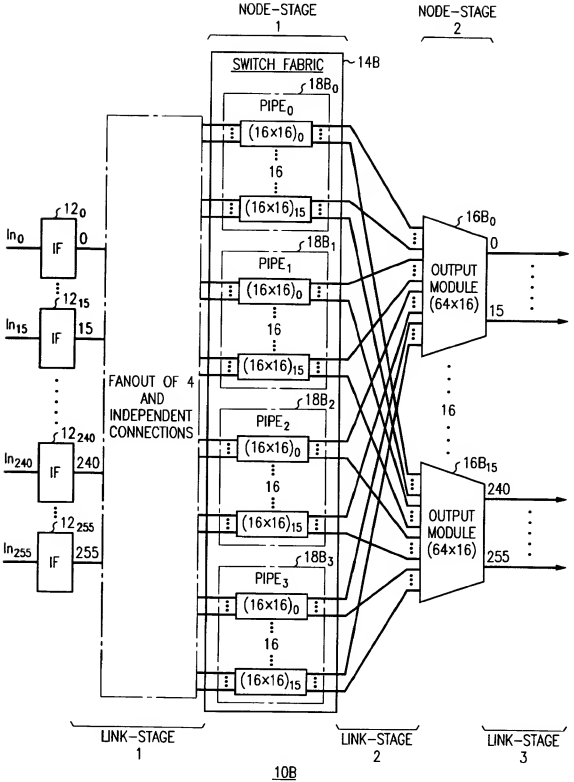


FIG. 18

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 720 413 A3

(12) EUROPEAN PATENT APPLICATION

(88) Date of publication A3:
06.05.1999 Bulletin 1999/18

(51) Int. Cl.[6]: **H04Q 11/04**, H04L 12/56

(43) Date of publication A2:
03.07.1996 Bulletin 1996/27

(21) Application number: **95309013.1**

(22) Date of filing: **12.12.1995**

(84) Designated Contracting States:
BE DE ES FR GB IT NL

(30) Priority: **30.12.1994 US 366704**
30.12.1994 US 367489
30.12.1994 US 366707
30.12.1994 US 366708

(71) Applicant: **AT&T Corp.**
New York, NY 10013-2412 (US)

(72) Inventors:
- **Cloonan, Thomas Jay**
Downers Grove, Illinois 60516 (US)
- **Richards, Gaylord Warner**
Lisle, Illinois 60532 (US)

(74) Representative:
Watts, Christopher Malcolm Kelway, Dr. et al
Lucent Technologies (UK) Ltd,
5 Mornington Road
Woodford Green Essex, IG8 0TU (GB)

(54) Terabit per second packet switch

(57) A physically realizable one terabit or more ATM packet switch 10A that has a large number of input interfaces connected to a single stage switching fabric 14A which is in turn connected to a number of output modules 16_0-16_m, according to the generic growable packet switch architecture. This ATM packet switch 10A is different from other growable packet switches in that it has a single stage switch fabric (14A or 14B) controlled by an out-of-band controller 20, yet it has significantly reduced complexity with respect to comparably sized electronic crossbar switches or their isomorphs. This ATM packet switch architecture is so flexible, it can be extended to provide variable length packets, circuit switched connections and fault tolerant redundant circuits using the same switch fabric and out-of-band controller architectures.

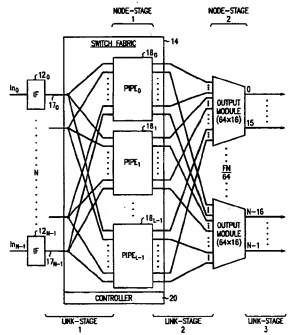


FIG. 3

EP 0 720 413 A3

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 95 30 9013

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
X	WO 93 03567 A (GPT LIMITED) 18 February 1993 * figure 2 * * figure 4 * * figure 5 * * figure 7 *	1-20,40	H04Q11/04 H04L12/56
A	* figure 20 * * page 9, line 11 - line 17 * * page 10, line 3 - line 7 * * page 10, line 21 - line 30 * * page 11, paragraph 5.1.1 * * page 12, line 15 - line 19 * * page 13, paragraph 5.1.2.2 * * page 13, paragraph 5.1.2.3 * -----	21-26,41	

TECHNICAL FIELDS SEARCHED (Int.Cl.6)

H04Q
H04L

~~The present search report has been drawn up for all claims~~

Place of search	Date of completion of the search	Examiner
THE HAGUE	18 November 1998	Lamadie, S

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

European Patent Office

Application Number
EP 95 30 9013

CLAIMS INCURRING FEES

The present European patent application comprised at the time of filing more than ten claims.

☐ Only part of the claims have been paid within the prescribed time limit. The present European search report has been drawn up for the first ten claims and for those claims for which claims fees have been paid, namely claim(s):

☐ No claims fees have been paid within the prescribed time limit. The present European search report has been drawn up for the first ten claims.

LACK OF UNITY OF INVENTION

The Search Division considers that the present European patent application does not comply with the requirements of unity of invention and relates to several inventions or groups of inventions, namely:

see sheet B

☐ All further search fees have been paid within the fixed time limit. The present European search report has been drawn up for all claims.

☐ Only part of the further search fees have been paid within the fixed time limit. The present European search report has been drawn up for those parts of the European patent application which relate to the inventions in respect of which search fees have been paid, namely claims:

☒ None of the further search fees have been paid within the fixed time limit. The present European search report has been drawn up for those parts of the European patent application which relate to the invention first mentioned in the claims, namely claims:

1-26,40,41

European Patent Office

LACK OF UNITY OF INVENTION
SHEET B

Application Number
EP 95 30 9013

The Search Division considers that the present European patent application does not comply with the requirements of unity of invention and relates to several inventions or groups of inventions, namely:

1. Claims: 1-26,40,41

   A packet switch for switching a telecommunication packet comprising plurality of pipes.

2. Claims: 27-39,60-67

   A controller with memory means for storing busy or idle status tables.

3. Claims: 42-47

   A plurality of crossbar switches for use in a telecommunication switch.

4. Claims: 48-50

   Single stage switch fabric having a single path between an input and an output, and a packet switch using the same.

5. Claims: 51-58

   A packet switch for circuit switched and packet switched communications comprising a plurality of circuit switched input and output modules.

6. Claims: 1,10,59

   Packet switch with spare unit for fault tolerance

7. Claim : 68

   Method for controlling paths for X ATM cells, comprising the step of dividing cells into two groups of m and X-m cells.

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 95 30 9013

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

18-11-1998

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9303567 A	18-02-1993	GB 2258366 A	03-02-1993
		EP 0551475 A	21-07-1993
		EP 0858192 A	12-08-1998
		JP 6501831 T	24-02-1994
		US 5459724 A	17-10-1995